Final Report for NASA Contract NNL07AA23C

ADJOINT-BASED METHODOLOGY FOR TIME-DEPENDENT OPTIMAL CONTROL (AMTOC)

Reporting Period: 02/07 – 02/12

Prepared by: **PI: Nail Yamaleev** ***(North Carolina A&T State University)***

Co-PI: Boris Diskin ***(National Institute of Aerospace)***

Co-I: Hiroaki Nishikawa ***(National Institute of Aerospace)***

The five-year project "Adjoint-based Methodology for Time-dependent Optimal Control (AMTOC)" has been awarded to NIA in February 2007. During the five years of this project, the AMTOC team developed an adjoint-based methodology for design and optimization of complex time-dependent flows, implemented AMTOC in a testbed environment, directly assisted in implementation of this methodology in the state-of-the-art NASA's unstructured CFD code FUN3D, and successfully demonstrated applications of this methodology to large-scale optimization of several supersonic and other aerodynamic systems, such as fighter jet, subsonic aircraft, rotorcraft, high-lift, wind-turbine, and flapping-wing configurations. In the course of this project, the AMTOC team has published 13 refereed journal articles, 21 refereed conference papers, and 2 NIA reports. The AMTOC team presented the results of this research at 36 international and national conferences, meeting and seminars, including International Conference on CFD, and numerous AIAA conferences and meetings. Selected publications that include the major results of the AMTOC project are enclosed in this report.

The major accomplishments and achievements of the AMTOC team for each Task with references to publications related to the corresponding task are presented below.

Task 1: Develop, implement, and validate AMTOC

- Adjoint-based methodology for optimization of time-dependent flows has been developed [J2, J8, C1, C16, C20].
- The methodology has been implemented in FUN3D, verified, and applied to large-scale design optimization of unsteady turbulent flows on dynamic unstructured grids and applied for a tilt rotor in a pitch-up maneuver into a forward flight regime and to a fighter jet configuration with simulated aeroelastic motions [J8, C16].
- The methodology has been extended to overset grids, verified, and applied to demonstrate aerodynamic optimizations of a wind turbine, a biologically-inspired flapping wing, and a complex helicopter configuration subject to trimming constraints [J2, C1].
- The developed methodology impacted several other projects including optimization of high-lift configurations with active flow control components [O1], design of rotors using the Navier-Stokes equations in a noninertial reference frame [O2], coupled CFD/sonic-boom adjoint methodology and its application to aircraft design [O3], and motivated studies of a new rigorous approach to grid adaptation based on error minimization. [C8, C10, C11].

Task 2: Develop analysis tools for unstructured finite-volume discretizations and apply them to analyze current and new highly accurate finite-volume discretizations proposed for implementation in FUN3D.

- Developed methodology for analysis of accuracy and robustness of unstructured finite-volume discretizations [J4, J7, J12, C2, C17]
- Analyzed the state-of-the-art finite-volume discretization methods and develop new methods to improve accuracy, efficiency, and robustness of cell-centered and node-centered finite-volume discretizations for inviscid fluxes [J3, J4, J11, C2, C9, C14]
- Analyzed the state-of-the-art finite-volume discretization methods and develop new methods to improve accuracy, efficiency, and robustness of cell-centered and node-centered finite-volume discretizations for viscous fluxes [J3, J7, J11, C2, C9, C17, C18]

- Developed consistent, accurate, and robust discretizations for agglomerated grids [J5, J7, J9]
- Developed a general methodology for constructing robust and accurate diffusion schemes [J6, J12, C13, C17] and extended this methodology to Navier-Stokes solvers [C5, C6, C15]

Task 3: Develop and implement general quantitative analysis tools for multigrid solutions on unstructured grids. Develop and implement efficient multigrid solvers for the discretized 3-D Navier-Stokes equations.

- Developed methodology for efficient and robust agglomeration multigrid for large-scale complex turbulent-flow simulations, implemented this methodology in FUN3D, and demonstrated significant convergence acceleration in large-scale aerodynamic simulations. [J5, J9, C7, C12, C17]
- Developed and assisted in the FUN3D implementation of general quantitative analysis methods for multigrid solutions [J5, J9, C12, C17]
- Analyzed iterative solution methods on complex grids [J4, C4, C17]

Task 4: Investigate, develop, and implement various strategies for making the AMTOC methodology more affordable in terms of memory and CPU time.

- Developed an optimal local-in-time methodology dramatically reducing the storage requirements for the adjoint solver in unsteady flow applications [J10, C19]
- Developed an efficient and accurate POD-based reduced-order model that provides dramatic reduction of the storage and CPU time required for solving arbitrary Mach number flows [J1,C3].

Task 5: Develop and implement a general methodology for control-volume agglomeration on unstructured grids, which is compatible with the FUN3D requirements and data structure.

- Developed a general, efficient, scalable, and robust advancing-front hierarchical agglomeration scheme [J9].
- Developed a general, efficient, scalable, and robust line-agglomeration methodology, applied it to practical anisotropic viscous grids, used the methodology in developed efficient agglomeration multigrid solvers in large-scale applications [J5, J7, J9, C7]

References

Journal Articles and Book Chapters

J1. N. K. Yamaleev and K. Pathak, “Nonlinear model reduction for unsteady discontinuous flows,” submitted to J. of Computational Physics.

J2. Nielsen E. J. and Diskin B., “Discrete Adjoint-Based Design Optimization of Unsteady Turbulent Flows on Dynamic Overset Unstructured Grids,” submitted to AIAA Journal

J3. Schwöppe A. and Diskin B., "Accuracy of the cell-centered grid metric in the TAU-code", to appear Notes on Numerical Fluid Mechanics and Multidisciplinary Design, Springer, NY

J4. Diskin B. and Thomas J. L., "Comparison of node-centered and cell-centered unstructured finite-volume discretizations: inviscid fluxes," AIAA Journal (2011), 49(4), pp. 836-854.
J5. Thomas J. L., Diskin B., and Nishikawa H., "A critical study of agglomerated multigrid methods for diffusion on highly stretched grids," Computers & Fluids (2011), 41(1), pp. 82-93.
J6. H. Nishikawa, "Robust and Accurate Viscous Discretization by Upwind Schemes - I: Basic Principle," Computers and Fluids, 49, pp. 62-86, 2011.
J7. Diskin B., Thomas J. L., Nielsen E. J., Nishikawa H., and White J. A., "Comparison of node-centered and cell-centered unstructured finite-volume discretizations: viscous fluxes," AIAA Journal (2010), 48(7), pp. 1326-1338.
J8. Nielsen E. J., Diskin B., and Yamaleev N. K., "Discrete adjoint-based design optimization of unsteady turbulent flows on dynamic unstructured grids," AIAA Journal (2010), 48(6), pp. 1195-1206
J9. Nishikawa H., Diskin B., and Thomas J. L., "A critical study of agglomerated multigrid methods for diffusion," AIAA Journal (2010), 48(4), pp. 839-847
J10. Yamaleev N. K., Diskin B., and Nielsen E. J., "Local-in-time adjoint-based methods for design optimization of unsteady flows," Journal of Computational Physics (2010), 229(14), pp. 5394-5407
J11. Diskin B. and Thomas J. L. "Notes on accuracy of finite-volume discretization schemes on irregular grids," Applied Numerical Mathematics(2010), 60(3), pp. 224-226
J12. H. Nishikawa, "A First-Order System Approach for Diffusion Equation. II. Unification of Advection and Diffusion," Journal of Computational Physics, 229, pp. 3989-4016, 2010.
J13. Thomas J. L., Diskin B., and Ramsey C. L., "Towards Verification of Unstructured-Grid Solvers," AIAA Journal(2008), 46(12), pp. 3070-3079

Referred Conference Articles

C1. Nielsen E. J. and Diskin B., "Discrete Adjoint-Based Design Optimization of Unsteady Turbulent Flows on Dynamic Overset Unstructured Grids," AIAA-2012-0554, Jan. 2012.
C2. Diskin B. and Thomas J. L., "Mesh effects on accuracy of finite-volume discretization schemes," invited paper, AIAA-2012-0609, Jan. 2012.
C3. Pathak, K., and Yamaleev, N., "POD-based reduced-order model for arbitrary Mach number flows," AIAA Paper 2011-3111, June, 2011.
C4. Diskin B. and Thomas J. L., "Convergence of defect-correction and multigrid iterations for inviscid flows," AIAA-2011-3235, June 2011
C5. H. Nishikawa, "New-Generation Hyperbolic Navier-Stokes Schemes: O(1/h) Speed-Up and Accurate Viscous/Heat Fluxes," AIAA Paper 2011-3043, June 2011.
C6. H. Nishikawa, "Two Ways to Extend Diffusion Schemes to Navier-Stokes Schemes: Gradient Formula or Upwind Flux," AIAA-2011-3044, June 2011
C7. Nishikawa H., Diskin B., and Thomas J. L., "Development and Application of Parallel Agglomerated Multigrid Method for Complex Geometries," AIAA-2011-3232, June 2011
C8. Diskin B. and Yamaleev N. K., "Grid Adaptation Using Adjoint-Based Error Minimization," AIAA-2011-3986, June 2011
C9. Schwöppe A. and Diskin B. "Genauigkeit der zell-zentrierten Netz-Metrik im TAU-Code" (in German), DLR STAB-Symposium, Berlin, Germany, Nov. 2010.

C10. Yamaleev N. K., Diskin B., and Pathak K., "Adjoint-Based Methodology for Anisotropic Mesh Adaptation," Proc. of the 6th International Conference on Computational Fluid Dynamics (ICCFD), St. Petersburg, Russia, July 2010.
C11. Yamaleev N. K., Diskin B., and Pathak K., "Error Minimization via Adjoint-Based Anisotropic Grid Adaptation" AIAA-2010-4436, June 2010.
C12. Nishikawa H., Diskin B., and Thomas J. L., "Development and Application of Agglomerated Multigrid Methods for Complex Geometries," AIAA-2010-4731, June 2010.
C13. H. Nishikawa, Beyond Interface Gradient: A General Principle for Constructing Diffusion Schemes, AIAA Paper 2010-5093, June 2010.
C14. Diskin B. and Thomas J. L., "Comparison of Cell-Centered and Node-Centered Unstructured Finite-Volume Discretizations: Inviscid Fluxes," AIAA-2010-1079, Jan. 2010
C15. H. Nishikawa, Towards Future Navier-Stokes Schemes: Uniform Accuracy, O(h) Time step, and Accurate Viscous/Heat Fluxes, AIAA Paper 2009-3648, June 2009
C16. Nielsen E. J., Diskin B., and Yamaleev N. K., "Discrete Adjoint-Based Design Optimization of Unsteady Turbulent Flows on Dynamic Unstructured Grids," AIAA-2009-3802, June 2009.
C17. Nishikawa H., Diskin B., and Thomas J. L., "A Critical Study of Agglomerated Multigrid Methods for Diffusion," AIAA-2009-4138, June 2009.
C18. Diskin B., Thomas J. L., Nielsen E. J., Nishikawa H., and White J. A., "Comparison of Node-Centered and Cell-Centered Unstructured Finite-Volume Discretizations. Part I: Viscous Fluxes," AIAA-2009-597, Jan. 2009.
C19. Yamaleev N. K., Diskin B., and Nielsen E. J., "Local-in-time adjoint-based method for design optimization of unsteady compressible flows," AIAA-2009-1169, Jan. 2009.
C20. Yamaleev N. K., Diskin B., and Nielsen E. J., "Adjoint-based Methodology for Time-Dependent Optimization," AIAA-2008-5857, Sept. 2008.
C21. Thomas J. L., Diskin B., and Rumsey C. L., "Towards Verification of Unstructured-Grid Solvers," AIAA-2008-666, Jan. 2008.

Technical Reports

T1. Diskin B. and Thomas J. L., "Accuracy of Gradient Reconstruction on Grids with High Aspect Ratio," NIA Report 2008-12.
T2. Diskin B. and Thomas J. L., "Accuracy Analysis for Mixed-Element Finite-Volume Discretization Schemes," NIA Report 2007-08.

Other Relevant Papers not authored, but with contributions from the team members

O1. Nielsen, E. and Jones, W., "Integrated Design of an Active Flow Control System Using a Time-Dependent Adjoint Method," Mathematical Modeling of Natural Phenomena, Vol. 6, No. 3, 2011, pp. 1195–1206.
O2. Nielsen, E., Lee-Rausch, E., and Jones, W., "Adjoint-Based Design of Rotors Using the Navier-Stokes Equations in a Noninertial Reference Frame," AIAA Journal of Aircraft, Vol. 47, No. 2, 2010, pp. 638–646.
O3. Rallabhandi, S. K., "Sonic Boom Adjoint Methodology and its Applications", AIAA-2011-3497, June 2011

Presentations:

1. H. Nishikawa, "Robust and Accurate Viscous Discretization by Hyperbolic Recipe," NIA CFD Seminar, National Institute of Aerospace, Hampton, VA, Nov. 1 2011.
2. B. Diskin, "Mesh quality effects on accuracy of node-centered finite-volume discretization schemes," NIA CFD Seminar, National Institute of Aerospace, Hampton, VA, Oct. 18, 2011
3. H. Nishikawa, "First-Order Hyperbolic System Method," NIA CFD Seminar, National Institute of Aerospace, Hampton, VA, Oct. 4 2011.
4. N. Yamaleev, "CPU Time and Storage Saving Techniques for Adjoint-based Optimization of Unsteady Flows," Computational Aerosciences Branch seminar, NASA LaRC, Hampton, VA, Sept. 13-14, 2011.
5. B. Diskin, "Advanced computational methods for complex aerodynamic flows: effective unstructured discretizations, adjoint-based optimization, and applications," NIA "Sandwich" Seminar, National Institute of Aerospace, Hampton, VA, Aug. 24, 2011
6. Yamaleev N. K., "Grid Adaptation Using Adjoint-Based Error Minimization," 20th AIAA Computational Fluid Dynamics Conference, Honolulu, Hawaii, June 2011
7. Diskin B., "Convergence of defect-correction and multigrid iterations for inviscid flows," 20th AIAA Computational Fluid Dynamics Conference, Honolulu, Hawaii, June 2011
8. H. Nishikawa, "New-Generation Hyperbolic Navier-Stokes Schemes: O(1/h) Speed-Up and Accurate Viscous/Heat Fluxes," 20th AIAA Computational Fluid Dynamics Conference, Honolulu, Hawaii, June 2011.
9. H. Nishikawa, "Two Ways to Extend Diffusion Schemes to Navier-Stokes Schemes: Gradient Formula or Upwind Flux," 20th AIAA Computational Fluid Dynamics Conference, Honolulu, Hawaii, June 2011
10. Nishikawa H., "Development and Application of Parallel Agglomerated Multigrid Method for Complex Geometries," 20th AIAA Computational Fluid Dynamics Conference, Honolulu, Hawaii, June 2011
11. B. Diskin, "Mesh quality effects on accuracy of node-centered finite-volume discretization schemes," DLR Institute of Aerodynamics and Flow Technology, Center of Computer Applications in Aerospace Science and Engineering, Braunschweig, Germany, May 27, 2011.
12. N. Yamaleev, "Adjoint-Based Methodology for Design Optimization of Unsteady Flows," Math. Seminar, Michigan Technological University, Houghton, Houghton, MI Feb. 2011.
13. Yamaleev N. K., "Adjoint-Based Methodology for Anisotropic Mesh Adaptation," 6th International Conference on Computational Fluid Dynamics, St. Petersburg, Russia, July 2010.
14. Yamaleev N. K., "Error Minimization via Adjoint-Based Anisotropic Grid Adaptation" 40th AIAA Fluid Dynamics Conference and Exhibit, Chicago, IL, June 2010.
15. Nishikawa H., "Development and Application of Agglomerated Multigrid Methods for Complex Geometries," 40th AIAA Fluid Dynamics Conference and Exhibit, Chicago, IL, June 2010.
16. H. Nishikawa, Beyond Interface Gradient: A General Principle for Constructing Diffusion Schemes, 40th AIAA Fluid Dynamics Conference and Exhibit, Chicago, IL, June 2010.

17. Diskin B., "Comparison of Cell-Centered and Node-Centered Unstructured Finite-Volume Discretizations: Inviscid Fluxes," 48th AIAA Aerospace Sciences Meeting, Orlando, FL, Jan. 2010.
18. B. Diskin, "Comparison of node-centered and cell-centered unstructured finite-volume discretizations: inviscid fluxes," DLR Institute of Aerodynamics and Flow Technology, Altenau, Germany, May 27, 2010.
19. Yamaleev N. K., Diskin B., NASA Supersonics NRA meeting, National Institute of Aerospace, Hampton, VA, Fed. 2010.
20. B. Diskin, "Computational tools for accuracy analysis of large-scale CFD codes," DLR Institute of Aerodynamics and Flow Technology, Center of Computer Applications in Aerospace Science and Engineering, Braunschweig, Germany, Nov. 4, 2009
21. B. Diskin, "Comparison of node-centered and cell-centered unstructured finite-volume discretizations for viscous fluxes," DLR Institute of Aerodynamics and Flow Technology, Center of Computer Applications in Aerospace Science and Engineering, Braunschweig, Germany, Nov. 5, 2009
22. B. Diskin, "Computational tools for quantitative analysis of multigrid solvers," DLR Institute of Aerodynamics and Flow Technology, Center of Computer Applications in Aerospace Science and Engineering, Braunschweig, Germany, Nov. 6, 2009
23. B. Diskin, "Computational analysis tools for large-scale computational fluid dynamics codes," Virginia Tech's Department of Aerospace and Ocean Engineering, Sept. 25, 2009.
24. Yamaleev N.K., "High-order CFD analysis and adjoint-based optimization of unsteady flows, CCMR Seminar, North Carolina A&T State University, Greensboro, NC, July 23, 2009.
25. Yamaleev N.K., "Discrete adjoint-based methodology for design optimization of unsteady flows," CFD Aerospace Seminar, University of Maryland at College Park, MD, June 9, 2009.
26. H. Nishikawa, Towards Future Navier-Stokes Schemes: Uniform Accuracy, O(h) Time step, and Accurate Viscous/Heat Fluxes, 19th AIAA Computational Fluid Dynamics Conference, San Antonio, TX, June 2009
27. Nishikawa H., "A Critical Study of Agglomerated Multigrid Methods for Diffusion," 19th AIAA Computational Fluid Dynamics Conference, San Antonio, TX, June 2009.
28. Yamaleev N. K., Diskin B., NASA Supersonics NRA meeting, National Institute of Aerospace, Hampton, VA, Fed. 2009.
29. Diskin B., "Comparison of Node-Centered and Cell-Centered Unstructured Finite-Volume Discretizations. Part I: Viscous Fluxes," 47th AIAA Aerospace Sciences Meeting Orlando, Florida, Jan. 2009.
30. Yamaleev N. K., "Locally Optimal Control/Design Strategies for Unsteady Compressible Flows," 47th AIAA Aerospace Sciences Meeting Orlando, Florida, Jan. 2009.
31. N. Yamaleev, "Adjoint-Based Methodology for Time-Dependent Aerodynamic Optimization," 5th South Eastern Atlantic Mathematical Sciences Workshop, Chapel Hill, NC, Nov. 2, 2008.
32. B. Diskin, "Computational Analysis Tools for Large-Scale Computational Fluid Dynamics Codes," 5th South Eastern Atlantic Mathematical Sciences Workshop, Chapel Hill, NC, Nov. 2, 2008

33. N. Yamaleev, “Adjoint-Based Methodology for Time-Dependent Aerodynamic Optimization,” Department of Mathematics, NC A&T State University, Oct. 2008.
34. N. Yamaleev, “Adjoint-based Methodology for Time-Dependent Optimization,” 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Victoria, BC, Canada, Sept. 2008.
35. Yamaleev N. K., Diskin B., NASA Supersonics NRA meeting, National Institute of Aerospace, Hampton, VA, Fed. 2008.
36. Diskin B., “Accuracy of Mixed-Element Finite Volume Discretization Schemes,” SIAM Conference on Computational Science and Engineering, Costa Mesa, CA, Feb 2007.

50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition
09 - 12 January 2012, Nashville, Tennessee
AIAA 2012-0554

Discrete Adjoint-Based Design for Unsteady Turbulent Flows On Dynamic Overset Unstructured Grids

Eric J. Nielsen*
Boris Diskin†

A discrete adjoint-based design methodology for unsteady turbulent flows on three-dimensional dynamic overset unstructured grids is formulated, implemented, and verified. The methodology supports both compressible and incompressible flows and is amenable to massively parallel computing environments. The approach provides a general framework for performing highly efficient and discretely consistent sensitivity analysis for problems involving arbitrary combinations of overset unstructured grids which may be static, undergoing rigid or deforming motions, or any combination thereof. General parent-child motions are also accommodated, and the accuracy of the implementation is established using an independent verification based on a complex-variable approach. The methodology is used to demonstrate aerodynamic optimizations of a wind turbine geometry, a biologically-inspired flapping wing, and a complex helicopter configuration subject to trimming constraints. The objective function for each problem is successfully reduced and all specified constraints are satisfied.

Nomenclature

$\mathbf{A}$	interpolation matrix
A, B	amplitudes of rotation in degrees
a, b, c, d	temporal coefficients
$\mathbf{C}$	$m_q \times 1$ vector of zeros and ones, indicator of time derivatives
$\mathbf{C}_s$	$m_s \times 1$ vector of zeros and ones, indicator of time derivatives at solve points
C	aerodynamic coefficient
C_L	lift coefficient
C_{M_x}, C_{M_y}	lateral and longitudinal moment coefficients
C_Q	torque coefficient
C_T	thrust coefficient
c	wing chord
$\mathbf{D}$	vector of design variables
E	total energy per unit volume
$\mathbf{F}$	flux vector
$\mathbf{F}_{inv}$, $\mathbf{F}_{visc}$	inviscid and viscous flux vectors
f,s	general functions
f_{obj}	objective function
g_1, g_2	explicit constraint functions
$\mathbf{G}$	grid operator
$\mathbf{I}$	projector operator
i	$\sqrt{-1}$

*Research Engineer, Computational AeroSciences Branch, NASA Langley Research Center, MS 128, Hampton, VA 23681, Senior Member AIAA.

†Associate Fellow, National Institute of Aerospace, 100 Exploration Way, Hampton, VA 23681, also Visiting Associate Professor, Department of Mechanical and Aerospace Engineering, University of Virginia, Charlottesville, VA 22904, Associate Fellow AIAA.

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

J	number of cost function components
$\mathbf{K}$	$m_x \times m_x$ linear elasticity coefficient matrix
L	Lagrangian function
m_d	size of vector $\mathbf{D}$
m_f	size of solution vector at fringe points
m_h	size of solution vector at hole points
m_q	size of solution vector $\mathbf{Q}$
m_s	size of solution vector at solve points
m_x	size of vector $\mathbf{X}$
N	number of time levels
n	time level
$\hat{\mathbf{n}}$	outward-pointing normal vector
$\mathbf{P}$	$m_h \times m_q$ pseudo-Laplacian matrix
p	pressure, also cost function exponent
$\mathbf{Q}$	$m_q \times 1$ vector of volume-averaged conserved variables
$\mathbf{q}$	$m_q \times 1$ vector of conserved variables
$\mathbf{R}$	$m_s \times 1$ vector of spatial undivided residuals
$\mathcal{R}$	$m_x \times m_x$ block-diagonal rotation matrix
R	3×3 rotation matrix
R_{GCL}	residual of the static geometric conservation law (GCL)
$\mathbf{R}_{GCL}$	$m_s \times 1$ vector of R_{GCL}
S	Control volume surface area
T	4×4 transform matrix
t	time
u, v, w	Cartesian components of velocity
$\mathbf{V}$	$m_q \times m_q$ diagonal matrix of cell volumes
V	control volume
$\mathbf{W}$	3×1 face velocity vector
$\mathbf{X}$	$m_x \times 1$ vector of grid coordinates
$\mathbf{x}$	3×1 position vector
x	independent variable
x, y, z	Cartesian coordinate directions
α	interpolation coefficient
β	scaling parameter for incompressible continuity equation
ε	perturbation
θ	angle of rotation, also blade pitch
θ_c	collective input
θ_{1c}	lateral cyclic input
θ_{1s}	longitudinal cyclic input
$\mathbf{\Lambda}$	$m_q \times 1$ flow-field adjoint variable
$\mathbf{\Lambda}_g$	$m_x \times 1$ grid adjoint variable
ρ	density
$\boldsymbol{\tau}$	$m_x \times 1$ translation vector
τ	3×1 translation vector
ψ	blade azimuth
ω	cost function component weight
ω_1, ω_2	frequencies of rotation in rad/s

Subscripts/Superscripts

c	child motion
f	fringe point
h	hole point
i, j, k, m, n	indices
in	quantity at initial conditions

nb	quantity at simply-connected neighbor
p	parent motion
s	solve point
x, y, z	axis of rotation
∞	quantity at free-stream conditions
$*$	target quantity
overbar	volume-averaged or time-averaged quantity, also complement of a vector

Symbols

Diag	diagonal matrix operator
$\circ$	Hadamard vector multiplication operator
$\odot$	extension of $\circ$ to a vector-matrix product

I. Introduction

As access to powerful high-performance computing resources has become more widespread in recent years, the use of high-fidelity physics-based simulation tools for analysis of complex aerodynamic flows has become increasingly routine. The availability and affordability of high-fidelity analysis tools has in turn stimulated an enormous body of research aimed at applying such tools to formal design optimization of complex aerospace configurations. A survey of the relevant literature shows that optimization methods based on the Euler and Reynolds-averaged Navier-Stokes equations have indeed gained a strong foothold in the design cycle for problems governed by steady flows.[1,2] Conversely, formal optimization methods for problems involving unsteady flow are also under development,[3–9] but in general are not as mature at the present time. This lag can be attributed at least in part to the increased computational cost typically associated with unsteady simulations.

For gradient-based optimization of problems involving many design variables, the ability to generate sensitivity information at a relatively low cost is critical. Unlike forward differentiation techniques such as finite differencing,[10] direct differentiation,[11] and complex-variable methods,[12] the adjoint approach performs sensitivity analysis at a cost comparable to that of a flow solution and independent of the number of design variables.[13] Efficient evaluation of sensitivities of an output with respect to all input parameters has led to numerous applications of adjoint-based methods in various areas of research and engineering. Some recent adjoint-based developments include a mathematically-rigorous approach to error estimation and mesh adaptation,[14] simultaneous design of shape and active flow control parameters for a high-lift configuration,[3] efficient methods for uncertainty quantification,[15] sonic boom optimization,[16] laminar flow control,[17] and many others.

Adjoint methods can be further classified into continuous and discrete variants, depending on the order in which the differentiation and discretization of the governing equations is performed. A discrete adjoint approach to sensitivity analysis is taken here. The methodology has been widely used for a broad class of optimization problems involving both steady and unsteady flows.[3,5,18–24] One of the advantages of the discrete adjoint approach is that the sensitivities computed by this method can be verified to machine precision by comparison with complex variable sensitivities.[12] The approach requires a complete linearization of the discrete governing equations with respect to both the flow-field variables and mesh coordinates. Strictly speaking, the adjoint approach for unsteady flows requires the evaluation of these linearizations at each physical time step. Therefore, the predominant challenge in extending a steady-state implementation to the unsteady regime is the development of an efficient infrastructure to store and access the entire forward solution as needed.

The analysis of vehicles experiencing large relative motion of vehicle components is often accomplished using overset discretizations. Design optimization for unsteady flows using such discretizations serves as the primary motivation for the current work. An implementation of the discrete adjoint approach for optimization of general three-dimensional unsteady turbulent flows on single-block unstructured grids is described in Refs. 3 and 5. Others have previously demonstrated adjoint-based capabilities for overset mesh discretizations; however, such works have been restricted to steady flows.[25–29] The methodology described here is intended for aerodynamic optimization of configurations characterized by large dynamic grid motions.

The primary contributions of this paper are the development, implementation, verification, and demonstration of an adjoint-based methodology for optimization and design of the most general unsteady aerodynamic flows. In the case of rotary wing flows, an optimization reported here involves a full helicopter configuration subject to trimming constraints and completes the series of studies addressing models of progressively higher fidelity. The previously considered models include actuator disk approaches,[30] noninertial formulations,[20] and dynamic grid formulations involving isolated rotors.[5] The generality of dynamic overset unstructured grid methods makes this methodology applicable to the most general flows occurring in a variety of practical computational fluid dynamics applications, e.g., store/stage separation, turbomachinery, wind turbine systems, rotary wing systems, biologically-inspired devices, and many others. Several diverse large-scale design applications are demonstrated in this paper.

The material is presented in the following order. The governing equations and some fundamental concepts of overset mesh systems are presented first. The approach taken to solve the flow equations is reviewed, followed by a derivation of the accompanying discrete adjoint equations. Details of the solution strategy are covered and the accuracy of the implementation for a very general dynamic motion case is verified using an independent approach based on complex variables. Finally, successful demonstrations of the design methodology are shown for a wind turbine geometry, a biologically-inspired flapping wing, and a realistic helicopter configuration. The appendix contains derivations for high-order temporal schemes.

II. Governing Equations

In this paper, the unsteady turbulent flow equations are used in both compressible and incompressible formulations. The primary distinction between these formulations is that the incompressible continuity equation does not have a time derivative term; all other (compressible and incompressible) equations do have time derivatives. For a simultaneous description of the unsteady compressible and incompressible Navier-Stokes equations, it is convenient to introduce an indicator of time derivative, $\mathbf{C}$, and a Hadamard vector multiplication operator.[31] The vector $\mathbf{C}$ is a logical vector composed of zeros and ones and has the same dimension as the residual vector. Ones correspond to equations with time derivatives, while zeros correspond to equations with no time derivatives. The logical complement to $\mathbf{C}$, $\bar{\mathbf{C}}$, is a vector of the same dimension in which zeros are replaced by ones and vice-versa. The Hadamard operator is denoted as $\circ$ and acts on two vectors of the same dimension, which are multiplied in an element-by-element fashion. The result of the Hadamard multiplication is a vector of the same dimension. The simultaneous description of the flow equations involves the Hadamard multiplication of the vector $\mathbf{C}$ with the vector of time derivatives. The resulting equations can be written in the following form for both moving and stationary control volumes:

$$\mathbf{C} \circ \frac{\partial}{\partial t} \int_V \mathbf{q} dV + \oint_{\partial V} (\mathbf{F}_{inv} - \mathbf{F}_{visc}) \cdot \hat{\mathbf{n}} dS = 0, \tag{1}$$

where V is the control volume bounded by the surface ∂V and $\hat{\mathbf{n}}$ is an outward-pointing unit normal. The vector $\mathbf{q}$ represents the conserved variables for mass, momentum, and energy, and the vectors $\mathbf{F}_{inv}$ and $\mathbf{F}_{visc}$ denote the inviscid and viscous fluxes, respectively.

For a moving control volume, the viscous flux is unchanged while the inviscid flux vector accounts for the difference in the fluxes due to the movement of control volume faces. Given an inviscid flux vector $\mathbf{F}$ on a static grid, the corresponding flux $\mathbf{F}_{inv}$ on a moving grid can be defined as $\mathbf{F}_{inv} = \mathbf{F} - (\mathbf{C} \circ \mathbf{q} + \bar{\mathbf{C}})(\mathbf{W} \cdot \hat{\mathbf{n}})$, where $\mathbf{W}$ is a local face velocity. In other words, $\mathbf{F}_{inv} = \mathbf{F} - \mathbf{q}(\mathbf{W} \cdot \hat{\mathbf{n}})$ for a conservation equation with a time derivative and $\mathbf{F}_{inv} = \mathbf{F} - (\mathbf{W} \cdot \hat{\mathbf{n}})$ for an equation without a time derivative.

By defining a volume-averaged quantity $\bar{\mathbf{q}}$ within each control volume,

$$\bar{\mathbf{q}} = \frac{1}{V} \int_V \mathbf{q} dV, \tag{2}$$

the conservation equations given by Eq. 1 take the form

$$\mathbf{C} \circ \frac{\partial (\bar{\mathbf{q}} V)}{\partial t} + \oint_{\partial V} (\mathbf{F}_{inv} - \mathbf{F}_{visc}) \cdot \hat{\mathbf{n}} dS = 0. \tag{3}$$

Here the conserved variables and inviscid flux vectors for compressible flows are defined as $\bar{\mathbf{q}} = [\rho, \rho u, \rho v, \rho w, E]^T$

and

$$\mathbf{F}_{inv} = \begin{bmatrix} \rho(u-W_x) \\ \rho u(u-W_x)+p \\ \rho v(u-W_x) \\ \rho w(u-W_x) \\ (E+p)(u-W_x)+W_x p \end{bmatrix} \hat{\mathbf{i}} + \begin{bmatrix} \rho(v-W_y) \\ \rho u(v-W_y) \\ \rho v(v-W_y)+p \\ \rho w(v-W_y) \\ (E+p)(v-W_y)+W_y p \end{bmatrix} \hat{\mathbf{j}} + \begin{bmatrix} \rho(w-W_z) \\ \rho u(w-W_z) \\ \rho v(w-W_z) \\ \rho w(w-W_z)+p \\ (E+p)(w-W_z)+W_z p \end{bmatrix} \hat{\mathbf{k}}, \quad (4)$$

and the perfect gas equation of state is assumed. The corresponding vectors for incompressible flows are $\bar{\mathbf{q}} = [p, u, v, w]^T$ and

$$\mathbf{F}_{inv} = \begin{bmatrix} \beta(u-W_x) \\ u(u-W_x)+p \\ v(u-W_x) \\ w(u-W_x) \end{bmatrix} \hat{\mathbf{i}} + \begin{bmatrix} \beta(v-W_y) \\ u(v-W_y) \\ v(v-W_y)+p \\ w(v-W_y) \end{bmatrix} \hat{\mathbf{j}} + \begin{bmatrix} \beta(w-W_z) \\ u(w-W_z) \\ v(w-W_z) \\ w(w-W_z)+p \end{bmatrix} \hat{\mathbf{k}}, \quad (5)$$

where β is a scaling parameter analogous to the artificial compressibility parameter.[32] Recall, however, that the incompressible continuity equation does not have a time derivative. The viscous flux vector $\mathbf{F}_{visc}$ is not explicitly shown here. For turbulent flows, the equations are closed with an appropriate turbulence model for the eddy viscosity.

The high-order (up to third-order) backward difference (BDF) discretizations for the time derivative of a function s are defined as

$$\frac{\partial s}{\partial t} = \frac{1}{\Delta t}\left(as^n + bs^{n-1} + cs^{n-2} + ds^{n-3}\right), \quad (6)$$

where n is a time level, and the coefficients are given in Table 1. The number after the BDF abbreviation indicates the order of the scheme. The coefficients listed for the BDF2opt scheme are a linear combination of the BDF2 and BDF3 coefficients taken from Refs. 33 and 34. The resulting scheme is second-order accurate but has a leading truncation error term less than that of the BDF2 scheme.

Using a BDF1 scheme, the discrete form of the flow equations at time level n is given as

$$\mathbf{C} \circ \frac{\bar{\mathbf{q}}^n V^n - \bar{\mathbf{q}}^{n-1} V^{n-1}}{\Delta t} + \mathbf{R}^n = 0, \quad (7)$$

where V^n and $\bar{\mathbf{q}}^n$ are a control volume and the corresponding solution vector at time level n and $\mathbf{R}^n$ is a vector of spatial undivided residuals approximating the flux term in Eq. 3. The first-order temporal scheme is chosen for the sake of simplicity; higher-order BDF schemes are used in practical computations and the demonstrations below. The Arbitrary Lagrangian-Eulerian (ALE)[35] node-centered finite-volume discretization of Eq. 3 used in the current work and described in Ref. 36 employs the following discrete formulation:

$$\mathbf{C} \circ \frac{\bar{\mathbf{q}}^n - \bar{\mathbf{q}}^{n-1}}{\Delta t} V^n + \mathbf{R}^n + R^n_{GCL}(\mathbf{C} \circ \bar{\mathbf{q}}^{n-1} + \beta \bar{\mathbf{C}}) = 0. \quad (8)$$

Here,

$$R^n_{GCL} = \oint_{\partial V} \mathbf{W}^n \cdot \hat{\mathbf{n}} dS, \quad (9)$$

where $\mathbf{W}^n$ is a vector of local face velocities at time level n. Note that substituting a spatially and temporally constant state vector, $\bar{\mathbf{q}}$, into Eq. 7 results in a geometric conservation law (GCL)[37]

$$\frac{V^n - V^{n-1}}{\Delta t} - R^n_{GCL} = 0 \quad (10)$$

for an equation with a time derivative and

$$-\beta R^n_{GCL} = 0 \quad (11)$$

for the incompressible continuity equation. Eq. 8 is obtained by subtracting the GCL residual, multiplied by $\bar{\mathbf{q}}^{n-1}$ for an equation with a time derivative, from Eq. 7.

III. Overset Grids

An overset grid formulation is characterized by the presence of two or more overlapping component grids. Each grid point and its corresponding control volume may be classified as one of four types based on the nature of the equation to be solved for that control volume. "Solve" points are points at which the discretized flow equations given by Eq. 8 are defined. "Fringe" points are points in overlap regions where interpolated data is specified in lieu of boundary conditions. The equations defined at fringe points are interpolation equations such that the solution at a fringe point, $\bar{\mathbf{q}}_f$, is defined as a linear combination of solution values at solve points, $\bar{\mathbf{q}}_s$:

$$\bar{\mathbf{q}}_f - \sum_k \alpha^k \bar{\mathbf{q}}_s^k = 0, \qquad \sum_k \alpha^k = 1. \tag{12}$$

Typically, the fringe point and the solve points appearing in Eq. 12 belong to different overlapping component grids. "Hole" points refer to points outside the computational domain, e.g., within the boundaries of a wing. In the current approach, the solution at hole points, $\bar{\mathbf{q}}_h$, is set to the average of the solution values at its simply-connected neighbors, $\bar{\mathbf{q}}_{nb}^j$. This averaging procedure is equivalent to a discrete pseudo-Laplacian, which is an elliptic operator:

$$\sum_j (\bar{\mathbf{q}}_h - \bar{\mathbf{q}}_{nb}^j) = 0, \tag{13}$$

where the hole point neighbors are identified by j. Finally, "orphan" points refer to grid points located within the computational domain for which neither the flow equations are imposed, nor can suitable points be found from which to interpolate solution information. In the current effort, the same pseudo-Laplacian procedure is defined for hole and orphan points, so that orphan points are not explicitly considered as a separate entity in the formulation to follow.

For dynamic grid motions, the character of each grid point may change as a function of time. It is preferable to have grid topologies such that the residuals of the governing equations at solve and fringe points do not depend on solution values at hole points. At a minimum, hole-point solutions should not contribute to residuals at solve and fringe points within the same time level. In practice, it can be difficult to prevent solutions at hole points from contributing to residuals at solve points through the time derivative; however, maximizing the extent of fringe regions and reducing the time step can help to alleviate this difficulty.

The domain-connectivity information required by the overset implementation is established using the software libraries described in Ref. 38. This methodology has been used extensively with the flow solver for performing analysis of multibody problems undergoing large relative motions.[30,36,39–45] Given the topology of each component grid, each grid point in the composite grid is determined to be a solve, fringe, hole or orphan point. This procedure is performed dynamically during the solution process as required by the grid motion. The mesh elements containing fringe points are established and the weighting coefficients required to interpolate data at such points are evaluated. For cases in which the grid motion is periodic, the user may choose to store the domain-connectivity information during the first cycle of motion for use in subsequent cycles. Once the interpolation coefficients are known, the complementary library described in Ref. 46 is used to determine the current solution at fringe points. The solution at hole and orphan points is determined based on user-supplied subroutines specifying the desired treatment at such locations. In the current approach, the pseudo-Laplacian given by Eq. 13 is used.

IV. Flow Solver

References 23, 36, and 47–49 describe the flow solver used in the current work. The code can be used to perform aerodynamic simulations across the speed range and an extensive list of options and solution algorithms is available for spatial and temporal discretizations on general static or dynamic mixed-element unstructured meshes which may or may not contain overset grid topologies.

In the current study, the spatial discretization uses a finite-volume approach in which the dependent variables are stored at the vertices of tetrahedral meshes. Inviscid fluxes at cell interfaces are computed using the upwind scheme of Roe,[50] and viscous fluxes are formed using an approach equivalent to a finite-element Galerkin procedure. The incompressible implementation is based on Refs. 49 and 51. For dynamic mesh cases, the mesh velocity terms are evaluated using backward differences consistent with the discrete

time derivative; this makes the spatial and GCL residuals dependent on grids at previous time levels. The eddy viscosity is modeled using the one-equation approach of Spalart and Allmaras.[52] The turbulence model is integrated all the way to the wall without the use of wall functions and is weakly coupled, i.e., solved separately from the mean flow equations at each time step. Scalability to thousands of processors is achieved through parallel domain decomposition, pre-processing, and solver mechanics. Post-processing operations such as the generation of isosurface and computational schlieren animations are also performed in parallel, avoiding the need for a single image of the mesh or solution at any time and ultimately yielding a highly efficient end-to-end parallel simulation paradigm. To date, this approach has been used to carry out computations on meshes containing as many as two billion points and twelve billion tetrahedral elements.[53]

To collectively describe equations and solutions defined at solve, fringe, and hole points, it is convenient to introduce corresponding projectors $\mathbf{I}_s^n$, $\mathbf{I}_f^n$, and $\mathbf{I}_h^n$ at time level n. These operators are rectangular matrices of respective dimensions $m_s \times m_q$, $m_f \times m_q$, and $m_h \times m_q$, and whose rows contain a single unity entry complemented by zeros. The values m_s, m_f, and m_h are the solution dimensions at all solve, fringe, and hole points, respectively, and $m_q = m_s + m_f + m_h$ is the solution dimension at all grid points. The projectors are used to extract solutions at grid points of a specific type: $\mathbf{Q}_s^n = \mathbf{I}_s^n \mathbf{Q}^n$, $\mathbf{Q}_f^n = \mathbf{I}_f^n \mathbf{Q}^n$, and $\mathbf{Q}_h^n = \mathbf{I}_h^n \mathbf{Q}^n$, where $\mathbf{Q}^n$ is the vector of solution values at all grid points and $\mathbf{Q}_s^n$, $\mathbf{Q}_f^n$, and $\mathbf{Q}_h^n$ are the vectors of solution values at solve, fringe, and hole points, respectively. Finally, note that the projector operators can vary in time.

The discrete form of the flow equations with a BDF1 scheme for the time derivative at time level n can be written as

$$\mathbf{C}_s^n \circ \mathbf{V}_s^n \circ \frac{\mathbf{Q}_s^n - \mathbf{I}_s^n \mathbf{Q}^{n-1}}{\Delta t} + \mathbf{R}^n + \left[\left(\mathbf{I}_s^n \mathbf{Q}^{n-1} \right) \circ \mathbf{C}_s^n + \beta \bar{\mathbf{C}}_s^n \right] \circ \mathbf{R}_{GCL}^n = 0. \tag{14}$$

In Eq. 14 and all discussions to follow, $\mathbf{R}^n$ and $\mathbf{R}_{GCL}^n$ are $m_s \times 1$ vectors that include residuals at solve points, $\mathbf{V}^n$ is an $m_q \times 1$ vector of control volumes for all equations at time level n, $\mathbf{V}_s^n = \mathbf{I}_s^n \mathbf{V}^n$ is a subset of $\mathbf{V}^n$ corresponding to solve points, $\mathbf{C}_s^n$ is an $m_s \times 1$ vector-indicator of a time derivative restricted to solve points at time level n, and $\bar{\mathbf{C}}_s^n$ is the complement of $\mathbf{C}_s^n$. Note that a solve point at time level n may or may not be a solve point at time level $n-1$.

The equations at fringe points are defined as

$$\mathbf{A}^n \mathbf{Q}^n = 0, \tag{15}$$

where $\mathbf{A}^n$ is the $m_f \times m_q$ matrix defining the interpolation of solution data from overset grid solutions at time level n as introduced in Eq. 12. The equations at hole points are defined as

$$\mathbf{P}^n \mathbf{Q}^n = 0, \tag{16}$$

where $\mathbf{P}^n$ is the $m_h \times m_q$ matrix of the pseudo-Laplacian given by Eq. 13.

The Jacobian of the implicit Eqs. 14, 15, and 16 at time level n is a 3×3 block matrix of the form

$$\begin{bmatrix} \frac{1}{\Delta t} \mathbf{Diag}(\mathbf{C}_s^n \circ \mathbf{V}_s^n) + \frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_s^n} & \frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_f^n} & \frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_h^n} \\ \mathbf{A}_s^n & \mathbf{A}_f^n & \mathbf{A}_h^n \\ \mathbf{P}_s^n & \mathbf{P}_f^n & \mathbf{P}_h^n \end{bmatrix}, \tag{17}$$

where $\mathbf{Diag}(\mathbf{C}_s^n \circ \mathbf{V}_s^n)$ is a diagonal $m_s \times m_s$ matrix with the vector $\mathbf{C}_s^n \circ \mathbf{V}_s^n$ on the main diagonal; $\mathbf{A}_f^n$ is an $m_f \times m_f$ diagonal matrix describing interpolation at fringe points; $\mathbf{A}_s^n$ and $\mathbf{A}_h^n$ are matrices with respective dimensions $m_f \times m_s$ and $m_f \times m_h$ describing interpolation from solve and hole points; and $\mathbf{P}_s^n$, $\mathbf{P}_f^n$, and $\mathbf{P}_h^n$ are matrices with respective dimensions $m_h \times m_s$, $m_h \times m_f$, and $m_h \times m_h$ describing contributions of solve, fringe, and hole points to the pseudo-Laplacian defined at hole points. Note that if the solution at hole points does not contribute to residuals at solve and fringe points within the same time level, then $\partial \mathbf{R}^n / \partial \mathbf{Q}_h^n = 0$, $\mathbf{A}_h^n = 0$, and the equations at hole points decouple from the equations at solve and fringe points.

V. Grid Equations

The general grid equations can be defined in the form

$$\mathbf{G}^n(\mathbf{X}, \mathbf{D}) = 0, \tag{18}$$

where the $m_x \times 1$ vector $\mathbf{X}$ represents the coordinates of the composite overset mesh (meshes at several time levels may be involved), $\mathbf{D}$ is the vector of design variables, and n denotes the time level and indicates that the grid operator may vary in time. Moreover, different grid operators $\mathbf{G}^n$ may be specified for different component grids. The specific formulations for different grid motions are introduced next.

V.A. Grids Undergoing Rigid Motion

For problems in which rigid mesh motion is required, the motion is generated by a 4×4 transform matrix, T, as outlined in Ref. 36. This transform matrix enables general translations and rotations of a grid according to the relation

$$\mathbf{x} = \mathbf{T}\mathbf{x}^0, \tag{19}$$

which moves a point from an initial position $\mathbf{x}^0 = (x^0, y^0, z^0)^T$ to its new position $\mathbf{x} = (x, y, z)^T$:

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & \tau_x \\ R_{21} & R_{22} & R_{23} & \tau_y \\ R_{31} & R_{32} & R_{33} & \tau_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^0 \\ y^0 \\ z^0 \\ 1 \end{bmatrix}. \tag{20}$$

In an expanded form, $\mathbf{x} = R\mathbf{x}^0 + \tau$. Here, the 3×3 matrix R defines a general rotation and the vector τ specifies a translation. The matrix $\mathbf{T}$ is generally time dependent. One useful feature of this approach is that multiple transformations telescope via matrix multiplication. This formulation is particularly attractive for composite parent-child body motion, in which the motion of one body is often specified relative to another. The reader is referred to the discussion in Ref. 36 for more details. For a rigid-motion formulation, the grid operator at time level n is defined as

$$\mathbf{G}^n(\mathbf{X}^n, \mathbf{X}^0, \mathbf{D}) \equiv \mathcal{R}^n\mathbf{X}^0 + \boldsymbol{\tau}^n - \mathbf{X}^n, \tag{21}$$

or in abbreviated notation,

$$\mathbf{G}^n(\mathbf{X}^n, \mathbf{X}^0, \mathbf{D}) \equiv \mathbf{T}^n\mathbf{X}^0 - \mathbf{X}^n. \tag{22}$$

Here, $\mathbf{X}^0$ and $\mathbf{X}^n$ are the grid vectors at the initial and n-th time levels, respectively; $\mathcal{R}^n$ is an $m_x \times m_x$ block-diagonal matrix with 3×3 blocks representing rotation and m_x being the size of vector $\mathbf{X}^n$; and $\boldsymbol{\tau}^n$ is an m_x-size translation vector. The matrix $\mathcal{R}^n$ and vector $\boldsymbol{\tau}^n$ may explicitly depend on $\mathbf{D}$.

V.B. Deforming Grids

The simplest example of a deforming grid simulation is a static grid undergoing deformations as a result of a shape optimization process. In this case, the grid is not time-dependent and is modeled as an elastic medium that obeys the elasticity relations of solid mechanics. An auxiliary system of linear partial differential equations (PDEs) is solved to determine the mesh coordinates after each shape update. Discretization of these PDEs yields a system of equations

$$\mathbf{K}\left(\mathbf{X} - \bar{\mathbf{X}}\right) = \mathbf{X}_{bound} - \bar{\mathbf{X}}_{bound}, \tag{23}$$

where $\mathbf{K}$ represents the elasticity coefficient matrix, $\mathbf{X}$ is the vector of grid coordinates being solved for, $\bar{\mathbf{X}}$ is the vector of coordinates of a reference grid, and $\mathbf{X}_{bound}$ and $\bar{\mathbf{X}}_{bound}$ are the vectors of corresponding boundary coordinates, complemented by zeros for all interior coordinates. The coefficients of the matrix $\mathbf{K}$ depend on $\bar{\mathbf{X}}$. The material properties of the system given by Eq. 23 are chosen based on either the local cell geometry or proximity to the surface and are invariant with respect to coordinate transformations. The system is solved using a preconditioned generalized minimal residual algorithm. For further details on the approach, see Refs. 19, 36, and 54.

For static grid deformation, the only grid operator used at all times is

$$\mathbf{G}(\mathbf{X}, \mathbf{D}) \equiv -\mathbf{K}\left(\mathbf{X} - \bar{\mathbf{X}}\right) + \mathbf{X}_{bound} - \bar{\mathbf{X}}_{bound}, \tag{24}$$

where $\mathbf{X}_{bound}$ may explicitly depend on $\mathbf{D}$, $\bar{\mathbf{X}}$ is an independent grid obtained either from a grid generator or from the previous optimization iteration, and $\bar{\mathbf{X}}_{bound}$ is the vector of corresponding boundary coordinates.

When time-dependent deforming grids are required, the rigid motion as described in the previous section is not valid. For small relative grid deformations, the linear elasticity equations given by Eq. 23 are solved at each time level with the matrix $\mathbf{K} = \mathbf{K}^0$ computed at the initial time level and fixed throughout the time evolution; $\mathbf{X}^n_{bound}$ includes the description of the current body positions. The grid operator at time level n is defined as

$$\mathbf{G}^n(\mathbf{X}^n, \mathbf{D}) \equiv -\mathbf{K}^0 \left(\mathbf{X}^n - \bar{\mathbf{X}}\right) + \mathbf{X}^n_{bound} - \bar{\mathbf{X}}_{bound}. \tag{25}$$

V.C. Parent-Child Motions

Large relative motions are described through parent-child relations, in which the collective motion of a child body is described as the product $\mathbf{T}_p\mathbf{T}_c$, where $\mathbf{T}_p$ is the collective parent transform matrix (which itself can be a chain of parent-child products) and $\mathbf{T}_c$ is the transform matrix describing the motion of the child with respect to the parent. In the current implementation, there is a one-to-one correspondence between moving bodies and component grids. Additional static grids may be associated with the non-inertial frame. Thus, a transform matrix describes not only the body motion, but may also describe the transformation of the corresponding grid. In general, a parent-child chain of motions can include an arbitrary combination of rigidly moving and deforming overset grids. If a component grid, $\mathbf{X}^n$, is designated as rigid, then all nodes of this grid undergo the same motion described as

$$\mathbf{G}^n(\mathbf{X}^n, \mathbf{X}^0, \mathbf{D}) \equiv -\mathbf{X}^n + \mathbf{T}_p\mathbf{T}_c\mathbf{X}^0. \tag{26}$$

If a component grid is designated as deforming, then the initial grid, $\mathbf{X}^0$, is either given,

$$\mathbf{G}^0(\mathbf{X}^0, \mathbf{D}) \equiv -\mathbf{X}^0 + \bar{\mathbf{X}}, \tag{27}$$

or computed from the elasticity equations, Eq. 25. The corresponding body surface undergoes the $\mathbf{T}_p\mathbf{T}_c$ motion, the external boundary and the initial (reference) grid undergo the $\mathbf{T}_p$ motion, and the grid at time level n, $\mathbf{X}^n$, satisfies the elasticity relations

$$\mathbf{G}^n(\mathbf{X}^n, \mathbf{X}^0, \mathbf{D}) \equiv -\mathbf{K}^n \left(\mathbf{X}^n - \mathbf{T}_p\mathbf{X}^0\right) + \mathbf{X}^n_{bound} - \mathbf{T}_p\mathbf{X}^0_{bound}. \tag{28}$$

Here, the matrix $\mathbf{K}^n$ is computed using the moved initial grid $\mathbf{T}_p\mathbf{X}^0$. Note that because of invariance of the material properties of the elasticity system, the following identity holds:

$$\mathbf{K}^n\mathbf{T}_p = \mathbf{T}_p\mathbf{K}^0. \tag{29}$$

In the current implementation, if any component grid is designated as deforming, then the entire composite grid is designated as deforming, and all component grids are treated as deforming, including those component grids that are in fact rigid. In this scenario, the external boundaries and the reference grid of a rigid component grid are moved with the collective motion of the corresponding body, $\mathbf{T}_p\mathbf{T}_c$, the boundary variations in Eq. 28 become zero, and the obtained grid, $\mathbf{X}^n$, is equivalent to the rigidly moving one, Eq. 26. If all component grids are labeled as either rigid or static, then the composite grid is designated rigid, and all grid points are moved according to Eq. 26.

VI. Cost Functions and Design Variables

The steady-state adjoint implementation described in Refs. 18–24 permits multiple objective functions and explicit constraints of the following form, each containing a summation of individual components:

$$f_i = \sum_{j=1}^{J_i} \omega_j (C_j - C_j^*)^{p_j}. \tag{30}$$

Here, ω_j represents a user-defined weighting factor, C_j is an aerodynamic coefficient such as the total drag or the pressure or viscous contributions to such quantities, the superscript $*$ indicates a user-defined target value of C_j, and p_j is a user-defined exponent. Targets are chosen to encourage beneficial changes in the design parameters and are typically far enough from the baseline values to avoid limiting potential improvements. The exponent values are chosen so that f_i is a convex functional, which is important for

convergence of gradient-based optimization. The user may specify computational boundaries to which each component function applies. The index i indicates a possibility of introducing several different cost functions or constraints, which may be useful if the user desires separate sensitivities, for example, for lift, drag, pitching moment, etc. The implementation also supports multipoint optimization.[20]

For the unsteady formulation, similar general cost functions f_i^n are defined at each time level n. The accumulated cost function f_i can be defined as a discrete sum over a certain time interval $[t_i^1, t_i^2]$:

$$f_i = \sum_{n=N_i^1}^{N_i^2} f_i^n, \tag{31}$$

where time levels N_i^1 and N_i^2 correspond to t_i^1 and t_i^2, respectively. The corresponding time integral is approximated as $f_i \Delta t$. The current study also introduces an additional cost function of the following form, which is based on the time-averaged value of an output:

$$f_i = \left[\left(\frac{1}{(N_i^2 - N_i^1 + 1)} \sum_{n=N_i^1}^{N_i^2} C_i^n \right) - C_i^* \right]^{p_i}. \tag{32}$$

The user supplies time intervals over which the cost functions are to be used.

There are three classes of design variables available in the current implementation. The first is composed of global parameters unrelated to the computational grid. These variables include parameters such as the free-stream Mach number and angle of attack. Such variables are particularly useful in verifying the implementation of the flow-field adjoint equations.

The second class of design variables provides general shape control of the configuration. The implementation allows the user to employ a geometric parameterization scheme of choice, provided the associated surface grid linearizations are available. For the examples in the current study, the grid parameterization approach described in Ref. 55 is used. This approach can be used to define general shape parameterizations of existing grids using a set of aircraft-centric design variables such as camber, thickness, shear, twist, and planform parameters at various locations on the geometry. The user also has the freedom to associate design variables to define more general parameters. In the event that multiple bodies of the same shape are to be designed — such as a set of rotor blades — the implementation allows for a single set of design variables to be used to simultaneously define such bodies. In this fashion, the shape of each body is constrained to be identical throughout the course of the design.

Finally, the third class of design variables governs any kinematics that may be present. The user may invoke simple translation and rotation functions native to the solver; in this case, basic parameters such as frequencies, amplitudes, directional vectors, and centers of rotation are available as design variables. Alternatively, more complicated kinematics and associated design variables may be supplied through a user-defined subroutine satisfying a standard interface. This interface is wrapped with a complex-variable perturbation scheme[12] to numerically evaluate the Jacobian of the specified kinematic motion which is required by the adjoint formulation to follow.

VII. Adjoint Equations

The goal of the design optimization problem for unsteady flows is to choose the design parameters $\mathbf{D}$ to minimize an objective function, $f_{obj} = f\Delta t$, where f is posed by Eqs. 31 or 32 and the subscript i is omitted. For the sake of clarity, the formulation to be presented here is based on a BDF1 scheme for the time derivative as introduced in Eq. 14. The derivation for higher order BDF schemes is similar and is presented in the appendix. Following the methodology described in Refs. 5 and 56, a Lagrangian function is defined as

$$\begin{aligned} L\left(\mathbf{D}, \mathbf{Q}, \mathbf{X}, \mathbf{\Lambda}, \mathbf{\Lambda}_g\right) &= f\Delta t + \left(\left[\mathbf{\Lambda}_g^0\right]^T \mathbf{G}^0 + \left[\mathbf{\Lambda}^0\right]^T \mathbf{R}^{in} \right) \Delta t \\ &+ \sum_{n=1}^{N} \left\{ \left[\mathbf{\Lambda}_g^n\right]^T \mathbf{G}^n + \left[\mathbf{\Lambda}_f^n\right]^T \left[\mathbf{A}^n \mathbf{Q}^n\right] + \left[\mathbf{\Lambda}_h^n\right]^T \left[\mathbf{P}^n \mathbf{Q}^n\right] \right. \\ &\left. + \left[\mathbf{\Lambda}_s^n\right]^T \left[\mathbf{C}_s^n \circ \mathbf{V}_s^n \circ \frac{\mathbf{Q}_s^n - \mathbf{I}_s^n \mathbf{Q}^{n-1}}{\Delta t} + \mathbf{R}^n + \left(\left(\mathbf{I}_s^n \mathbf{Q}^{n-1}\right) \circ \mathbf{C}_s^n + \beta \bar{\mathbf{C}}_s^n \right) \circ \mathbf{R}_{GCL}^n \right] \right\} \Delta t \end{aligned} \tag{33}$$

Here, $\mathbf{\Lambda}_s^n$, $\mathbf{\Lambda}_f^n$, $\mathbf{\Lambda}_h^n$ and $\mathbf{\Lambda}_g^n$ are $m_s \times 1$, $m_f \times 1$, $m_h \times 1$, and $m_x \times 1$ vectors of Lagrange multipliers associated with the solve, fringe, hole, and grid equations, respectively; $[\mathbf{\Lambda}^n]^T = \left[[\mathbf{\Lambda}_s^n]^T, \left[\mathbf{\Lambda}_f^n\right]^T, [\mathbf{\Lambda}_h^n]^T\right]$; $\mathbf{\Lambda}_s^n = \mathbf{I}_s^n \mathbf{\Lambda}^n$, $\mathbf{\Lambda}_f^n = \mathbf{I}_f^n \mathbf{\Lambda}^n$, and $\mathbf{\Lambda}_h^n = \mathbf{I}_h^n \mathbf{\Lambda}^n$; and $\mathbf{R}^{in} = 0$ represents the initial conditions. A typical form of the initial conditions is $\mathbf{R}^{in} \equiv \mathbf{V}^0 \circ \left(\mathbf{Q}_\infty - \mathbf{Q}^0\right)$, where $\mathbf{Q}_\infty$ is the free-stream solution; other forms, such as a steady-state initial solution, are also possible.

The Lagrangian given by Eq. 33 is differentiated with respect to $\mathbf{D}$, assuming that $\mathbf{V}^n$ depends on $\mathbf{X}^n$; $\mathbf{G}^n$ depends on $\mathbf{X}^n$, $\mathbf{X}^0$, and $\mathbf{D}$; $\mathbf{R}^n$ depends on $\mathbf{Q}^n$, $\mathbf{X}^n$, $\mathbf{X}^{n-1}$, and $\mathbf{D}$; $\mathbf{R}_{GCL}^n$ depends on $\mathbf{X}^n$, $\mathbf{X}^{n-1}$, and $\mathbf{D}$; $\mathbf{A}^n$ depends on $\mathbf{X}^n$; $\mathbf{G}^0$ depends on $\mathbf{X}^0$ and $\mathbf{D}$; $\mathbf{R}^{in}$ depends on $\mathbf{Q}^0$, $\mathbf{X}^0$, and $\mathbf{D}$; and $\mathbf{P}^n$, $\mathbf{C}_s^n$, $\bar{\mathbf{C}}_s^n$, $\mathbf{I}_s^n$, $\mathbf{I}_f^n$, and $\mathbf{I}_h^n$ are independent of grid coordinates, solutions, and design parameters.

Regrouping terms to collect the coefficients of $\partial \mathbf{Q}^n / \partial \mathbf{D}$ and equating those coefficients to zero yields the adjoint equations:

$$
\begin{aligned}
S: \quad & \tfrac{1}{\Delta t}\mathbf{C}_s^n \circ \mathbf{V}_s^n \circ \mathbf{\Lambda}_s^n + \left[\tfrac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_s^n}\right]^T \mathbf{\Lambda}_s^n + [\mathbf{A}_s^n]^T \mathbf{\Lambda}_f^n + [\mathbf{P}_s^n]^T \mathbf{\Lambda}_h^n = \\
& - \left[\tfrac{\partial f}{\partial \mathbf{Q}_s^n}\right]^T - \mathbf{I}_s^n \left[\mathbf{I}_s^{n+1}\right]^T \left[\mathbf{C}_s^{n+1} \circ \left(-\tfrac{1}{\Delta t}\mathbf{V}_s^{n+1} + \mathbf{R}_{GCL}^{n+1}\right) \circ \mathbf{\Lambda}_s^{n+1}\right], \\
F: \quad & \left[\tfrac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_f^n}\right]^T \mathbf{\Lambda}_s^n + \left[\mathbf{A}_f^n\right]^T \mathbf{\Lambda}_f^n + \left[\mathbf{P}_f^n\right]^T \mathbf{\Lambda}_h^n = \\
& - \left[\tfrac{\partial f}{\partial \mathbf{Q}_f^n}\right]^T - \mathbf{I}_f^n \left[\mathbf{I}_s^{n+1}\right]^T \left[\mathbf{C}_s^{n+1} \circ \left(-\tfrac{1}{\Delta t}\mathbf{V}_s^{n+1} + \mathbf{R}_{GCL}^{n+1}\right) \circ \mathbf{\Lambda}_s^{n+1}\right], \\
H: \quad & \left[\tfrac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_h^n}\right]^T \mathbf{\Lambda}_s^n + [\mathbf{A}_h^n]^T \mathbf{\Lambda}_f^n + [\mathbf{P}_h^n]^T \mathbf{\Lambda}_h^n = \\
& - \left[\tfrac{\partial f}{\partial \mathbf{Q}_h^n}\right]^T - \mathbf{I}_h^n \left[\mathbf{I}_s^{n+1}\right]^T \left[\mathbf{C}_s^{n+1} \circ \left(-\tfrac{1}{\Delta t}\mathbf{V}_s^{n+1} + \mathbf{R}_{GCL}^{n+1}\right) \circ \mathbf{\Lambda}_s^{n+1}\right], \qquad for\ 1 \le n \le N; \\
& \left[\tfrac{\partial \mathbf{R}^{in}}{\partial \mathbf{Q}^0}\right]^T \mathbf{\Lambda}^0 = - \left[\tfrac{\partial f}{\partial \mathbf{Q}^0}\right]^T - \left[\mathbf{I}_s^1\right]^T \left[\mathbf{C}_s^1 \circ \left(-\tfrac{1}{\Delta t}\mathbf{V}_s^1 + \mathbf{R}_{GCL}^1\right) \circ \mathbf{\Lambda}_s^1\right], \qquad for\ n = 0,
\end{aligned}
\tag{34}
$$

where $\mathbf{\Lambda}_s^{N+1} = 0$. The preceding letters indicate the type of points at which the equations are defined; S, F, and H correspond to solve, fringe, and hole points, respectively. Collecting the coefficients of $\partial \mathbf{X}^n / \partial \mathbf{D}$ and equating those coefficients to zero in a similar fashion yields the grid adjoint equations:

$$
\begin{aligned}
& - \left[\tfrac{\partial \mathbf{G}^n}{\partial \mathbf{X}^n}\right]^T \mathbf{\Lambda}_g^n = \left[\left(\mathbf{C}_s^n \circ \tfrac{\mathbf{Q}_s^n - \mathbf{I}_s^n \mathbf{Q}^{n-1}}{\Delta t}\right) \odot \tfrac{\partial \mathbf{V}_s^n}{\partial \mathbf{X}^n}\right]^T \mathbf{\Lambda}_s^n + \left[\tfrac{\partial (\mathbf{A}^n \mathbf{Q}^n)}{\partial \mathbf{X}^n}\right]^T \mathbf{\Lambda}_f^n \\
& + \sum_{k=0}^{1} \left[\tfrac{\partial \mathbf{R}^{n+k}}{\partial \mathbf{X}^n} + \left(\left(\mathbf{I}_s^{n+k} \mathbf{Q}^{n+k-1}\right) \circ \mathbf{C}_s^{n+k} + \beta \bar{\mathbf{C}}_s^{n+k}\right) \odot \tfrac{\partial \mathbf{R}_{GCL}^{n+k}}{\partial \mathbf{X}^n}\right]^T \mathbf{\Lambda}_s^{n+k} + \left[\tfrac{\partial f}{\partial \mathbf{X}^n}\right]^T, \qquad for\ 1 \le n \le N; \\
& - \left[\tfrac{\partial \mathbf{G}^0}{\partial \mathbf{X}^0}\right]^T \mathbf{\Lambda}_g^0 = \sum_{n=1}^{N} \left[\tfrac{\partial \mathbf{G}^n}{\partial \mathbf{X}^0}\right]^T \mathbf{\Lambda}_g^n + \left[\tfrac{\partial \mathbf{R}^{in}}{\partial \mathbf{X}^0}\right]^T \mathbf{\Lambda}^0 \\
& + \sum_{k=1}^{1} \left[\tfrac{\partial \mathbf{R}^k}{\partial \mathbf{X}^0} + \left(\left(\mathbf{I}_s^k \mathbf{Q}^{k-1}\right) \circ \mathbf{C}_s^k + \beta \bar{\mathbf{C}}_s^k\right) \odot \tfrac{\partial \mathbf{R}_{GCL}^k}{\partial \mathbf{X}^0}\right]^T \mathbf{\Lambda}_s^k + \left[\tfrac{\partial f}{\partial \mathbf{X}^0}\right]^T, \qquad for\ n = 0.
\end{aligned}
\tag{35}
$$

Here, $\partial f / \partial \mathbf{X}^n$ is a $1 \times m_x$ row vector, $\partial \mathbf{G}^n / \partial \mathbf{X}^n$ is an $m_x \times m_x$ matrix, $\partial \mathbf{V}_s^n / \partial \mathbf{X}^n$, $\partial \mathbf{R}^n / \partial \mathbf{X}^m$, and $\partial \mathbf{R}_{GCL}^n / \partial \mathbf{X}^m$ are $m_s \times m_x$ matrices, $\partial (\mathbf{A}^n \mathbf{Q}^n) / \partial \mathbf{X}^n$ is an $m_f \times m_x$ matrix, and $\partial \mathbf{R}^{in} / \partial \mathbf{X}^0$ is an $m_q \times m_x$ matrix. The operation $\odot$ is an extension of the Hadamard multiplication to a product between an $m_s \times 1$ vector and an $m_s \times m$ matrix, where the second matrix dimension, m, is arbitrary. The operation indicates that the vector multiplies the columns of the matrix in an element-by-element fashion resulting in a new $m_s \times m$ matrix.

When considering the linearization of $\mathbf{A}^n$, the domain-connectivity information is assumed to be fixed. That is, the weighting coefficients represented by this matrix are considered functions of the mesh coordinates; however, the interpolating elements are considered constant so that the hole-cutting and domain-connectivity algorithms need not be linearized.

With Lagrangian multipliers satisfying equations Eqs. 34 and 35, the sensitivity derivatives are calculated

as follows:

$$\frac{\partial L}{\partial \mathbf{D}} = \frac{\partial f}{\partial \mathbf{D}}\Delta t + \sum_{n=1}^{N} \left[\mathbf{\Lambda}_g^n\right]^T \frac{\partial \mathbf{G}^n}{\partial \mathbf{D}}\Delta t + \sum_{n=1}^{N} \left[\mathbf{\Lambda}_s^n\right]^T \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{D}} + \left(\left(\mathbf{I}_s^n \mathbf{Q}^{n-1}\right) \circ \mathbf{C}_s^n + \beta \bar{\mathbf{C}}_s^n\right) \odot \frac{\partial \mathbf{R}_{GCL}^n}{\partial \mathbf{D}}\right] \Delta t$$
$$+ \left(\left[\mathbf{\Lambda}_g^0\right]^T \frac{\partial \mathbf{G}^0}{\partial \mathbf{D}} + \left[\mathbf{\Lambda}^0\right]^T \left[\frac{\partial \mathbf{R}^{in}}{\partial \mathbf{D}}\right]\right) \Delta t, \tag{36}$$

where $\partial L/\partial \mathbf{D}$ and $\partial f/\partial \mathbf{D}$ are $1 \times m_d$ row vectors, $\partial \mathbf{G}^n/\partial \mathbf{D}$ is an $m_x \times m_d$ matrix, $\partial \mathbf{R}^n/\partial \mathbf{D}$ and $\partial \mathbf{R}_{GCL}^n/\partial \mathbf{D}$ are $m_s \times m_d$ matrices, and $\partial \mathbf{R}^{in}/\partial \mathbf{D}$ is an $m_q \times m_d$ matrix.

To facilitate the solution of Eqs. 34 and 35, the values of $\mathbf{X}^n$, $\partial \mathbf{X}^n/\partial t$, and $\mathbf{Q}^n$ are stored to disk at the conclusion of each physical time step of the flow solution using a strategy designed to minimize file system overhead. The approach is based on a massively parallel paradigm in which each processor writes to its own unformatted direct-access file at each time step. The data writes are buffered using an asynchronous paradigm so that execution of floating point operations for the subsequent time step may proceed simultaneously. This approach is described and evaluated in Ref. 3 and has been found to scale well to several thousand processors using a parallel file system. Rather than recompute the domain-connectivity information during the adjoint solution procedure, a similar I/O paradigm has been implemented to efficiently store this information to disk, although the size of this data is typically an order of magnitude less than the flow-field data. During the solution of Eqs. 34 and 35, data is loaded from disk using a similar paradigm but in reverse, such that data required for the solution at time level $n-1$ is pre-loaded during the computations for time level n.

VIII. Iterative Solution of Equations at Each Time Level

When solving the flow equations, the value of $\mathbf{Q}^{n-1}$ is taken to be an initial approximation for $\mathbf{Q}^n$. The solution of Eqs. 14, 15, and 16 at time level n is obtained through the following iterations, which exploit the form of the Jacobian matrix given by Eq. 17:

$$\begin{aligned} F: \quad & \mathbf{A}_f^n \Delta \mathbf{Q}_f^{n,m} = -\left[\mathbf{A}_s^n \mathbf{Q}_s^{n,m} + \mathbf{A}_f^n \mathbf{Q}_f^{n,m} + \mathbf{A}_h^n \mathbf{Q}_h^{n,m}\right], \\ & \mathbf{Q}_f^{n,m+1} = \mathbf{Q}_f^{n,m} + \Delta \mathbf{Q}_f^{n,m}, \end{aligned} \tag{37}$$

$$\begin{aligned} S: \quad & \left[\tfrac{1}{\Delta\tau}\mathbf{Diag}(\mathbf{V}_s^n) + \tfrac{1}{\Delta t}\mathbf{Diag}(\mathbf{C}_s^n \circ \mathbf{V}_s^n) + \tfrac{\partial \hat{\mathbf{R}}^{n,m}}{\partial \mathbf{Q}_s^n}\right] \Delta \mathbf{Q}_s^{n,m} = \\ & -\left[\mathbf{C}_s^n \circ \tfrac{\mathbf{Q}_s^{n,m} - \mathbf{I}_s^n \mathbf{Q}^{n-1}}{\Delta t} \circ \mathbf{V}_s^n + \mathbf{R}^{n,m} + \left(\left(\mathbf{I}_s^n Q^{n-1}\right) \circ \mathbf{C}_s^n + \beta \bar{\mathbf{C}}_s^n\right) \circ \mathbf{R}_{GCL}^n\right], \\ & \mathbf{Q}_s^{n,m+1} = \mathbf{Q}_s^{n,m} + \Delta \mathbf{Q}_s^{n,m}, \end{aligned} \tag{38}$$

$$\begin{aligned} H: \quad & \mathbf{P}^n \Delta \mathbf{Q}_h^{n,m} = -\left[\mathbf{P}_s^n \mathbf{Q}_s^{n,m+1} + \mathbf{P}_f^n \mathbf{Q}_f^{n,m+1} + \mathbf{P}_h^n \mathbf{Q}_h^{n,m}\right], \\ & \mathbf{Q}_h^{n,m+1} = \mathbf{Q}_h^{n,m} + \Delta \mathbf{Q}_h^{n,m}. \end{aligned} \tag{39}$$

Here, the second superscript m is the iteration count, $\mathbf{R}^{n,m}$ is the spatial non-linear residual computed for the most recent solution that involves $\mathbf{Q}_f^{n,m+1}$, $\Delta\tau$ is a pseudo-time step, and $\partial \hat{\mathbf{R}}^{n,m}/\partial \mathbf{Q}_s^n$ is the Jacobian of a first-order spatial discretization.

At each iteration, Eq. 37 is solved exactly because $\mathbf{A}_f^n$ is a diagonal matrix, and the fringe solutions are updated first. An approximate solution of the linear system of equations (Eq. 38) is obtained through several iterations of a multicolor Gauss-Seidel point-iterative scheme, followed by a solution update for $\mathbf{Q}_s^{n,m+1}$. Finally, Eq. 39 is relaxed and solutions at hole points are updated. The convergence rate of the solution at hole points is typically the slowest; relaxation of the pseudo-Laplacian operator is known for poor convergence behavior. If the solution at hole points is decoupled, then its value may be updated only once after the solution at flow and fringe points has been converged.

The adjoint equations are solved backwards in time. The solution procedure outlined here is based on the single-grid implementation which has been previously verified for turbulent flows on three-dimensional unstructured grids undergoing general dynamic motions.[5] The iterative solution of the adjoint equations given by Eq. 34 at time level n is performed in precisely the reverse order as the iterations given by Eqs. 37-39:

$$
\begin{aligned}
H: \quad & [\mathbf{P}_h^n]^T \Delta\mathbf{\Lambda}_h^{n,m} = -\left[\tfrac{\partial f}{\partial \mathbf{Q}_h^n}\right]^T - \mathbf{I}_h^n \left[\mathbf{I}_s^{n+1}\right]^T \left[\mathbf{C}_s^{n+1} \circ \left(-\tfrac{1}{\Delta t}\mathbf{V}_s^{n+1} + \mathbf{R}_{GCL}^{n+1}\right) \circ \mathbf{\Lambda}_s^{n+1}\right] \\
& - [\mathbf{P}_h^n]^T \mathbf{\Lambda}_h^{n,m} - \left[\tfrac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_h^n}\right]^T \mathbf{\Lambda}_s^{n,m} - [\mathbf{A}_h^n]^T \mathbf{\Lambda}_f^{n,m}, \\
& \mathbf{\Lambda}_h^{n,m+1} = \mathbf{\Lambda}_h^{n,m} + \Delta\mathbf{\Lambda}_h^{n,m},
\end{aligned} \tag{40}
$$

$$
\begin{aligned}
S: \quad & \left[\tfrac{1}{\Delta\tau}\mathbf{Diag}(\mathbf{V}_s^n) + \tfrac{1}{\Delta t}\mathbf{Diag}(\mathbf{C}_s^n \circ \mathbf{V}_s^n) + \tfrac{\partial \hat{\mathbf{R}}^{n,m}}{\partial \mathbf{Q}_s^n}\right] \Delta\mathbf{\Lambda}_s^{n,m} = \\
& - \left[\tfrac{\partial f}{\partial \mathbf{Q}_s^n}\right]^T - \mathbf{I}_s^n \left[\mathbf{I}_s^{n+1}\right]^T \left[\mathbf{C}_s^{n+1} \circ \left(-\tfrac{1}{\Delta t}\mathbf{V}_s^{n+1} + \mathbf{R}_{GCL}^{n+1}\right) \circ \mathbf{\Lambda}_s^{n+1}\right] \\
& - \tfrac{1}{\Delta t}\mathbf{C}_s^n \circ \mathbf{V}_s^n \circ \mathbf{\Lambda}_s^{n,m} - \left[\tfrac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_s^n}\right]^T \mathbf{\Lambda}_s^{n,m} - [\mathbf{A}_s^n]^T \mathbf{\Lambda}_f^{n,m} - [\mathbf{P}_s^n]^T \mathbf{\Lambda}_h^{n,m+1} \\
& \mathbf{\Lambda}_s^{n,m+1} = \mathbf{\Lambda}_s^{n,m} + \Delta\mathbf{\Lambda}_s^{n,m},
\end{aligned} \tag{41}
$$

$$
\begin{aligned}
F: \quad & \left[\mathbf{A}_f^n\right]^T \Delta\mathbf{\Lambda}_f^{n,m} = -\left[\tfrac{\partial f}{\partial \mathbf{Q}_f^n}\right]^T - \mathbf{I}_f^n \left[\mathbf{I}_s^{n+1}\right]^T \left[\mathbf{C}_s^{n+1} \circ \left(-\tfrac{1}{\Delta t}\mathbf{V}_s^{n+1} + \mathbf{R}_{GCL}^{n+1}\right) \circ \mathbf{\Lambda}_s^{n+1}\right] \\
& - \left[\tfrac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_f^n}\right]^T \mathbf{\Lambda}_s^{n,m+1} - \left[\mathbf{A}_f^n\right]^T \mathbf{\Lambda}_f^{n,m} + \left[\mathbf{P}_f^n\right]^T \mathbf{\Lambda}_h^{n,m+1}, \\
& \mathbf{\Lambda}_f^{n,m+1} = \mathbf{\Lambda}_f^{n,m} + \Delta\mathbf{\Lambda}_f^{n,m}.
\end{aligned} \tag{42}
$$

Solutions for the grid adjoint equations are obtained through relaxation of Eq. 35.

IX. Verification of Adjoint Implementation

To verify the accuracy of the implementation, comparisons are made with results generated through an independent approach based on the use of complex variables. This approach was originally suggested in Refs. 12 and 57, and was first applied to a Navier-Stokes solver in Ref. 58. Using this formulation, an expression for the derivative of a real-valued function $f(x)$ may be found by expanding the function in a complex-valued Taylor series, using an imaginary perturbation $i\varepsilon$:

$$
\frac{\partial f}{\partial x} = \frac{Im[f(x+i\varepsilon)]}{\varepsilon} + O(\varepsilon^2). \tag{43}
$$

The primary advantage of this method is that true second-order accuracy may be obtained by selecting step sizes without concern for subtractive cancellation errors typically present in real-valued Frechet derivatives. Through the use of an automated scripting procedure outlined in Ref. 59, this capability can be immediately recovered at any time for the baseline flow solver. For computations using this method, the imaginary step size has been chosen to be 10^{-50}, which highlights the robustness of the complex-variable approach. For each verification test, all equation sets are converged to machine precision for both the complex-variable and adjoint approaches. Since the package described in Ref. 46 cannot directly accommodate complex-valued grids and solutions, the integer-valued donor and receptor information is instead transferred to the solver, which performs the requisite complex-valued donor weight computations and solution interpolations. This procedure has been verified to produce identical real components as compared to the routines internal to the package of Ref. 46.

The test case used to verify the accuracy of the implementation is based on the rotorcraft configuration shown in Fig. 1. The conventional rotorcraft definition for the azimuth angle ψ is also shown in the figure. The fuselage is described by a component mesh consisting of 88,001 nodes and 505,437 tetrahedral elements. Each of the four rotor blades is modeled using a component grid containing 103,296 nodes and 601,459 tetrahedral elements. The entire configuration is combined with a background grid consisting of 50,156 nodes and 285,587 tetrahedral elements to yield a composite mesh system with 551,341 nodes and 3,196,860 tetrahedral elements.

A very general combination of forced motions is applied to the configuration as follows. The fuselage mesh is subjected to a rigid fixed-rate rotational and translational motion in the starboard direction. The motion of each rotor blade is treated as a child of the fuselage motion, and consists of an additional rigid fixed-rate rotation in the azimuthal direction. Each blade is also subjected to a final child motion consisting of a forced vertical flapping that is modeled as a 1° oscillatory rotation about the rotor hub with a two-per-revolution

frequency, and is accommodated with the deforming mesh mechanics. The background mesh is held fixed in inertial space. The overall motion of the configuration is shown in Fig. 2, while the vertical extent of the blade tip motion due to flapping is shown in Fig. 3. In summary, the composite motion is a family of four generations, occurring in the following ancestral order from oldest to youngest: inertial reference frame, fuselage motion, azimuthal blade motion, and flapping blade motion.

For the verification of the compressible implementation, the free-stream Mach number is 0.1 and the Reynolds number is 4.2 million based on the blade tip speed and chord, and fully turbulent flow is assumed. A similarly scaled Reynolds number of 3.1 million is used for the incompressible verification. The angle of attack is 2°, and the advance ratio is 0.12. The physical time step corresponds to 1° of rotation in the azimuthal direction. All of the computations are performed using 128 processors.

Sensitivity derivatives of the lift coefficient for the entire vehicle after five physical time steps are computed using the discrete adjoint and complex-variable approaches. Although the coarse spatial resolution and brief duration of the simulation are not sufficient to resolve the flow physics of the problem, they are adequate to evaluate the discrete consistency of the implementation. Table 2 shows the compressible flow sensitivity derivatives with respect to angle of attack, variables characterizing the rigid-body motions, and parameters describing the blade and fuselage shape. Results are shown for all of the temporal BDF schemes discussed in Section II and the appendix. Analogous results for the incompressible formulation are shown in Table 3. The results from the discrete adjoint and complex-variable approaches are in very good agreement for all cases; non-matching digits in the sensitivities are underlined.

X. Large-Scale Test Cases

To evaluate the proposed design methodology, aerodynamic optimizations are performed using three large-scale test cases. The goal is solely to demonstrate the ability of the implementation to successfully reduce each of the stated objective functions while satisfying any constraints present. While details pertaining to the underlying flow physics clearly may be of interest in each case, investigations of that nature are considered beyond the scope of the current effort and are not explored here.

For each case shown below, the spatial and temporal grid resolutions have been chosen based on a suitable compromise between solution accuracy and computational efficiency. Each optimization is performed on an SGI ICE system using dual-socket hex-core nodes with Intel Xeon X5670 cores in a fully-dense configuration. A single additional node is allocated for serial execution of the dynamic hole-cutting library. The computational environment also includes a Lustre-based parallel file system,[60] and computational statistics include any disk I/O time required to read or write the complete flow-field solution.

As described above, the implementation supports very general motions including the use of deforming bodies. However, physical models typically responsible for such effects — such as structural models — generally are a strong function of the aerodynamics and require a formal coupling procedure. While the flow solver used in the current study can accommodate such models, the adjoint formulation does not account for such effects at this time. Therefore, to evaluate the current methodology, all large-scale simulations described here rely on forced motions. Development of a more general adjoint formulation required for coupling aerodynamics with other disciplinary models is relegated to future work.

X.A. NREL Phase VI Wind Turbine

The first test case is based on the NREL Phase VI wind turbine described in Ref. 61. The geometry is a two-blade upwind configuration with a nacelle and tower. The grid system used here has been developed in Ref. 43. The component grid for each blade consists of 4,510,177 nodes and 26,574,786 tetrahedral elements, and a separate component grid containing the nacelle and tower geometries consists of 971,059 nodes and 5,716,227 tetrahedral elements. The background mesh consists of 4,776,082 nodes and 28,278,639 tetrahedral elements. The resulting composite mesh system contains 14,767,495 nodes and 87,144,438 tetrahedral elements. Views of the configuration and surface meshes are shown in Fig. 4.

The simulation is fully turbulent and is performed using the incompressible form of the governing equations. Standard sea-level conditions are used with a free-stream velocity of fifteen meters per second aligned with the axis of rotation. The radius of the blades is 5.029 meters and the system rotates at a speed of seventy-two RPM. The BDF2opt time integration scheme is used with 100 subiterations and a physical time step corresponding to 1° of blade rotation. Solutions are run for 720 time steps or two complete revolutions

of the blades. The torque profile for the baseline geometry is shown as the solid line in Fig. 5. After a series of initial transients, the solution quickly settles into a quasi-steady state behavior. The mean value of the torque coefficient $\bar{C}_Q$ measured over the second revolution is 0.00130. An isosurface of the Q criterion[62] is included in Fig. 6.

The goal of the current test case is to maximize the torque acting on the turbine by altering the blade geometry. The objective function is based on torque values $\hat{C}_Q$, which do not include the nondimensionalization using the reference geometry, and is posed as a discrete summation of the intermediate torque value minus a constant target value over the second revolution:

$$f_{obj} = \sum_{n=361}^{720} (\hat{C}_Q^n - 2.0)^2 \Delta t. \tag{44}$$

The target value of 2.0 has been chosen based on the initial $\hat{C}_Q$ profile. The objective function could also be formulated in terms of nondimensional torque values; in this case, the target value should be rescaled accordingly. There are a total of 76 design variables as shown in Fig. 7. These include seven twist values located at various stations along the span of the blade as well as twenty-one thickness and forty-eight camber variables distributed across the blade planform. Thinning of the blade is not allowed.

The optimization is performed using 240 computational nodes or a total of 2,880 processing cores. In this environment, individual flow-field and adjoint solutions require 6.5 and 6 hours of wall-clock time, respectively. Approximately 950 gigabytes of disk space are required to store a complete flow-field solution and its associated domain connectivity data. The package described in Ref. 63 is used to perform the optimization.

The convergence history for the optimization is shown in Fig. 8. The objective function has been reduced from its initial value of 69.4 to a final value of 58.7. The final profile for the torque coefficient is included as the dashed line in Fig. 5. The mean value $\bar{C}_Q$ measured over the second revolution is 0.00159, an increase of 22% over the baseline value. Cross-sections of the baseline and final blade geometries are shown in Fig. 9. The optimization has increased the thickness across much of the span, while also increasing the negative camber in the trailing edge region.

The optimization procedure for the current test case required nine flow solutions and eight adjoint solutions, for a total of 307,000 CPU hours or 4.5 days of wall-clock time. Although not done for the wind turbine demonstration, practical constraints such as root-bending moment or thrust constraints are straightforward to incorporate as shown in Section X.C.

X.B. Biologically-Inspired Flapping Wing

The next test case is based on a simple wing configuration undergoing a complex kinematic motion inspired by insects such as the Hawkmoth *manduca sexta*.[64] Such concepts are receiving considerable attention in applications to micro air vehicles.[65] The geometry consists of a rectangular flat plate with semi-circular leading and trailing edges and an aspect ratio of 3.33. The mesh system used for this example has been generated using the approach outlined in Ref. 66. The component mesh containing the wing geometry consists of 3,016,149 nodes and 17,642,078 tetrahedral elements. The background mesh containing the plane of symmetry and outer boundaries consists of 5,339,195 nodes and 31,446,042 tetrahedral elements, yielding a composite mesh with 8,355,344 nodes and 49,088,120 tetrahedral elements. A nearfield view of the wing surface mesh is shown in Fig. 10.

The baseline wing is offset 1.33 chord lengths from the plane of symmetry and is assumed to be operating in quiescent conditions. The imposed motion is achieved through the user-defined kinematics interface described above. Here, time-varying angles describing rotations about the x-, y-, and z-axes are specified in the following general form:

$$\begin{aligned} \theta_x &= A_x[\cos(\omega_{1x}t) - 1] + B_x \sin(\omega_{2x}t), \\ \theta_y &= A_y[\cos(\omega_{1y}t) - 1] + B_y \sin(\omega_{2y}t), \\ \theta_z &= A_z[\cos(\omega_{1z}t) - 1] + B_z \sin(\omega_{2z}t), \end{aligned} \tag{45}$$

where the amplitudes and frequencies are specified by the user. These angles are used to construct a series of rotation matrices of the form given by Eq. 20. These matrices are then multiplied together to form the final rotation matrix used to specify the current wing position.

In the current example, the baseline motion is a superposition of two oscillatory rotations, each occurring at 26 Hz. The first rotation is a sweeping motion that rotates the wing $\pm 60^\circ$ about its root chord line. The second rotation is a feathering motion that rotates the wing $\pm 45^\circ$ about its leading edge. The net effect of this composite motion is a thrust force in the direction from trailing edge to leading edge. Several snapshots of the wing undergoing a period of the baseline motion are shown in Fig. 11.

The Reynolds number based on the wing chord and maximum tip speed is 1,280. The governing equations are the incompressible laminar Navier-Stokes equations. The BDF2opt time integration scheme is used with fifty subiterations and a physical time step corresponding to 250 steps per period of the baseline motion. Each simulation is run for 1,250 time steps and is performed using 160 computational nodes or a total of 1,920 processing cores. Approximately 850 gigabytes of disk space are required to store a complete flow-field solution and its associated domain connectivity data. Individual flow-field and adjoint solutions require roughly four and three hours of wall-clock time, respectively. The baseline thrust profile exhibits a two-per-cycle periodic behavior as shown by the solid line in Fig. 12. The mean value of the thrust coefficient C_T measured over the final period is 0.127.

The goal of the two test cases presented here is to maximize the thrust coefficient over the final 250 time steps by optimizing the fifteen design parameters describing the kinematic motion of the wing, namely the frequencies, amplitudes, and coordinates of the center of rotation for the composite motion described above. Both of the optimizations have been performed using the package described in Ref. 67. The first test case uses an objective function based on a target thrust distribution:

$$f_{obj} = \sum_{n=1,001}^{1,250} (C_T^n - 5.0)^2 \Delta t. \tag{46}$$

The second test case uses an objective function which aims to match a single target value for the time-averaged value of thrust:

$$f_{obj} = \left[\left(\frac{1}{250} \sum_{n=1,001}^{1,250} C_T^n \right) - 5.0 \right]^2 \Delta t. \tag{47}$$

In each case, the target value of 5.0 has been chosen based on the initial thrust profile shown in Fig. 12. Although not shown, physical constraints such as power constraints can also be incorporated in a straightforward fashion.

The convergence history for the objective function based on a target distribution is shown by the square symbols in Fig. 13. The value has been steadily reduced from 729 to 706 over ten design cycles. Inspection of the final values of the design variables shown in Table 4 reveals moderate changes to all parameters. The final thrust profile is included as the dashed line in Fig. 12. The optimization has not only increased the magnitude of the peaks, but has also altered the frequency content such that three peaks now occur within the interval used to define the objective function. The mean value of the thrust coefficient over the final 250 time steps is 0.207, a 63% increase over the baseline value. For this test, the optimizer requested twenty-two flow solutions and ten adjoint solutions, requiring approximately 227,000 CPU hours or five days of wall-clock time.

The results based on the time-average objective function are included in Fig. 12 as the dash-dot line. As in the previous case, the frequency of the signal has been altered to yield three peaks within the objective function interval. The mean value of the thrust coefficient over the final 250 time steps has been increased to 0.265, a 109% increase over the baseline value. The objective function history is plotted in Fig. 13, where it can be seen that the value has been reduced from 2.92 to 2.75 over eight design cycles. Here, the optimizer requested twenty-five flow solutions and eight adjoint solutions, requiring 238,000 CPU hours or just over five days of wall-clock time.

It should be noted that a series of shape optimizations were also attempted for the current test problem, but are not presented here. A total of eighty-eight shape parameters describing the twist, shear, thickness, and camber of the wing were used. In general, any shape modification yielding a thrust improvement over one half of the period was seen to be equally detrimental to performance during the opposite half, as each wing surface alternates between pressure and suction conditions. Other forms of shape modification such as planform effects could prove beneficial, although such changes have not been explored here.

X.C. UH-60A Blackhawk Helicopter

The final test case is based on the UH-60A Blackhawk helicopter configuration.[68] Extensive analysis of this configuration has previously been performed using the solver employed in the current study.[39] The composite grid system used here consists of four identical blade component grids and a single component grid containing the fuselage and outer extent of the computational domain. Each of the blade grids consists of 1,266,525 nodes and 7,476,818 tetrahedral elements, while the fuselage grid contains 4,196,841 nodes and 24,735,227 tetrahedral elements. This results in a composite grid system consisting of 9,262,941 nodes and 54,642,499 tetrahedral elements. The surface mesh for the configuration is shown in Fig. 14.

The governing equations are the compressible Reynolds-averaged Navier-Stokes equations. The simulation is based on a forward flight condition with a blade tip Mach number equal to 0.6378 and a Reynolds number of 7.3 million based on the blade tip chord. The advance ratio is 0.37 and the angle of attack is 0°. The rotor blades are subjected to a time-dependent pitching motion that is modeled as a child of the azimuthal rotation and is governed by a sinusoidal variation based on collective and cyclic control inputs:

$$\theta = \theta_c + \theta_{1c} cos\psi + \theta_{1s} sin\psi. \tag{48}$$

Here, θ is the current blade pitch setting, ψ is the current azimuth position for the blade, θ_c represents the collective control input, and θ_{1c} and θ_{1s} are the lateral and longitudinal cyclic control inputs, respectively. All three control inputs are set to 0° at the baseline condition; i.e., the vehicle is initially untrimmed.

The BDF2opt time integration scheme is used with fifteen subiterations and a physical time step corresponding to 1° of rotor rotation. The simulation is run for two rotor revolutions using 160 computational nodes or a total of 1,920 processing cores. In this environment, a single execution of the flow and adjoint solvers requires two and three hours of wall-clock time, respectively. Approximately 650 gigabytes of disk space are required to store a complete flow-field solution and its associated domain connectivity data.

Figure 15 shows an isosurface of the Q criterion[62] after two rotor revolutions. The vortices emanating from each blade tip and other surfaces of the vehicle are clearly visible. Profiles of the baseline lift and lateral and longitudinal moment cofficients are shown as the solid lines in Figs. 16-18. The values quickly establish a four-per-revolution periodic behavior after 180° of blade rotation. The mean value of the lift coefficient over the second rotor revolution is 0.023. The untrimmed flight condition is clearly evident in the nonzero mean values for the two moment coefficients.

The objective for the current test case is to maximize the lift acting on the vehicle while satisfying explicit constraints on the lateral and longitudinal moments such that the final result is a trimmed flight condition. The design variables consist of 64 shape parameters describing the rotor blades, including an 8x4 matrix of 32 thickness variables and 32 camber variables as shown in Fig. 19. While the camber is allowed to increase or decrease, no thinning of the blade is allowed. In addition, Eq. 48 and its relationship to the blade pitch transform matrix are also linearized, allowing the control variables θ_c, θ_{1c}, and θ_{1s} to also be used as design variables. These control angles are allowed to vary as much as ±7°. Note that parameters describing geometric changes to the fuselage could also be applied; however, without guidance for practical constraints on such changes, such variables are not used here.

The objective function to be minimized is based on the time-averaged value of the lift coefficient over the second rotor revolution:

$$f_{obj} = \left[\left(\frac{1}{360} \sum_{n=361}^{720} C_L^n \right) - 2.0 \right]^2 \Delta t. \tag{49}$$

The target value of 2.0 has been chosen based on the initial lift profile. The explicit constraints on the two moment coefficients are also based on time-averaged values over the same interval:

$$g_1 = \frac{1}{360} \sum_{n=361}^{720} C_{M_x}^n \Delta t \tag{50}$$

$$g_2 = \frac{1}{360} \sum_{n=361}^{720} C_{M_y}^n \Delta t. \tag{51}$$

The constraints are considered satisfied if $g_1 = g_2 = 0$, within a feasibility tolerance of ±0.0001. The optimization is performed using the package described in Ref. 63. Note that the treatment of the moment

constraints requires two additional adjoint solutions to compute the associated gradient vectors. These additional solutions are obtained simultaneously with the adjoint computation for the lift objective using the procedure outlined in Ref. 24 to accommodate multiple right-hand side vectors in Eqs. 34-36.

X.C.1. Design Results

Figure 20 shows the convergence of the objective function and constraints after three design cycles. The optimization procedure quickly locates a feasible region in the design space based on the two moment constraints and the value of the objective function is successfully reduced. The final unsteady lift profile is included as the dashed line in Fig. 16. The mean value has been substantially increased to a value of 0.103. The final unsteady profiles for the lateral and longitudinal moment coefficients are included as the dashed lines in Figs. 17 and 18, respectively. Each of the new profiles has the desired zero mean value, indicating that the final design is trimmed for level flight within the requested tolerance.

Based on the spanwise blade stations noted in Fig. 19, cross-sections of the initial and final blade geometries are shown in Fig. 21. The shape changes are confined to the aft sections of the outer portion of the blade, where the camber has been increased. The final value of the collective input θ_c is 6.71°, while the final values for the cyclic inputs θ_{1c} and θ_{1s} are 2.58° and -7.00°, respectively. The entire optimization procedure requiring four flow solutions and four adjoint solutions took approximately 20 hours of wall-clock time, or 38,400 CPU hours.

X.C.2. Interpretation of the Adjoint Solution

Typical qualitative features of unsteady adjoint solutions are shown in Fig. 22 for the objective function given by Eq. 49. The figure depicts centerline contours of the adjoint solution for the energy equation at time level $n = 420$. The contours represent the instantaneous sensitivity of the objective function to a source term applied to the energy equation at each point in the domain. Similar to steady-flow objective functions based on surface integrals,[69–72] the time-averaged value of the lift is particularly sensitive to information propagating along the stagnation streamline and impacting the nose of the fuselage. In addition, Fig. 22 highlights several features emanating from the rotor blades as they pass through the cutting plane. These features are loosely analogous to unsteady flow phenomena such as vortex sheets and tip vortices commonly seen in forward solutions for rotor flows as shown in Fig. 15. However, unlike the forward problem, the features shown in the adjoint solution propagate in the upstream direction as the adjoint system is integrated in reverse physical time, indicating the sensitivity of the objective function to disturbances upstream.

In design optimization, the adjoint solutions are combined with the linearizations of the residual operators with respect to design variables to yield sensitivity derivatives. Alternatively, the adjoint solutions may be combined with local residuals to provide rigorous error estimates or with (local estimates of) the truncation errors to guide mesh adaptation. Although these applications are not the focus of the current work, adjoint-based adaptation methodologies[14] offer many compelling advantages over traditional feature-based mesh adaptation techniques which fail to identify important regions such as those containing the upstream features highlighted in Fig. 22.

XI. Summary and Future Work

A general verified methodology for adjoint-based design optimization of unsteady turbulent flows on dynamic overset unstructured mesh systems has been presented. The formulation is valid for compressible and incompressible forms of the Reynolds-averaged Navier-Stokes equations. The implementation is amenable to massively parallel computing environments and has been verified through the use of an independent technique based on a complex-variable formulation. Several large-scale optimizations have been demonstrated for complex flowfields involving a wind turbine configuration, a flapping wing, and a realistic helicopter geometry subject to trimming constraints. The objective functions have been successfully reduced in each case and all constraints present have been satisfied.

Although the demonstrated methodology provides a practical approach to optimization of general unsteady aerodynamic flows, a wide range of research topics remains to be explored. Locally optimal,[73] reduced-order model,[74] and checkpointing[15] techniques offer the potential to greatly reduce storage requirements. Multi-fidelity optimization algorithms[75] should be exploited where possible to reduce dependence on high-fidelity simulations. Convergence acceleration techniques[76] can clearly have a direct impact on

computational cost. Simultaneous adjoint-based error estimation and mesh adaptation approaches[14] are very attractive in establishing rigorous gridding requirements and eliminating user interaction. Extension of adjoint-based methods to multidisciplinary optimization beyond the scope of computational fluid dynamics is essential for making significant impacts on the current paradigm for design of aerospace vehicles and other areas of applications. Finally, advancements in the fields of computer science, software development, and high-performance computing must continue to be leveraged to the greatest extent possible.

XII. Acknowledgments

The authors wish to thank Robert Biedron of NASA Langley Research Center and Nail Yamaleev of North Carolina A&T State University for many helpful conversations pertaining to the current work. The enabling contributions of Elizabeth Lee-Rausch and William Jones of NASA Langley Research Center in the areas of grid generation and parameterization are very appreciated. The work has also greatly benefitted from the high-performance computing research conducted by Dana Hammond of NASA Langley Research Center. Discussions with David Darmofal, Qiqi Wang, and Josh Krakos of the Massachusetts Institute of Technology (MIT) and Yuping Sun of Vestas Wind Systems were very useful. Ralph Noack of Penn State University is acknowledged for his support and assistance in using his overset grid libraries. Thanks are due to C. Eric Lynch of NAVAIR-Patuxent River for providing the mesh used for the wind turbine example. The authors are grateful to John Moore of MIT and the staff of the Air Force Research Laboratory at Wright-Patterson Air Force Base for providing the mesh and problem definition for the flapping wing case. The first author is supported by the Subsonic Rotary Wing element of the Fundamental Aeronautics Project within the NASA Aeronautics Research Mission Directorate. The second author acknowledges the support from NASA through contract #NNL07AA23C and cooperative agreement #NNL09AA00A.

Appendix: Adjoint Equations for Higher-Order BDF Schemes

Discrete conservation laws employing high order temporal BDF schemes as introduced in Eq. 6 are defined as

$$\begin{aligned}&\mathbf{C}_s^n \circ \left[a\tfrac{\mathbf{Q}_s^n - \mathbf{I}_s^n\mathbf{Q}^{n-1}}{\Delta t} \circ \mathbf{V}_s^n + c\tfrac{\mathbf{I}_s^n\mathbf{Q}^{n-2} - \mathbf{I}_s^n\mathbf{Q}^{n-1}}{\Delta t} \circ \mathbf{I}_s^n\mathbf{V}^{n-2} + d\tfrac{\mathbf{I}_s^n\mathbf{Q}^{n-3} - \mathbf{I}_s^n\mathbf{Q}^{n-1}}{\Delta t} \circ \mathbf{I}_s^n\mathbf{V}^{n-3}\right] \\ &+\mathbf{R}^n + \left((\mathbf{I}_s^n\mathbf{Q}^{n-1}) \circ \mathbf{C}_s^n + \beta\bar{\mathbf{C}}_s^n\right) \circ \mathbf{R}_{GCL}^n = 0.\end{aligned} \tag{52}$$

Proceeding as before, the Lagrangian can be written as

$$\begin{aligned}&L\left(\mathbf{D},\mathbf{Q},\mathbf{X},\mathbf{\Lambda},\mathbf{\Lambda}_g\right) = f\Delta t + \sum_{n=1}^{N}\left[\mathbf{\Lambda}_g^n\right]^T\mathbf{G}^n\Delta t \\ &+\sum_{n=1}^{N}\Big\{[\mathbf{C}_s^n \circ \mathbf{\Lambda}_s^n]^T\left[a\tfrac{\mathbf{Q}_s^n - \mathbf{I}_s^n\mathbf{Q}^{n-1}}{\Delta t} \circ \mathbf{V}_s^n + c\tfrac{\mathbf{I}_s^n\mathbf{Q}^{n-2} - \mathbf{I}_s^n\mathbf{Q}^{n-1}}{\Delta t} \circ (\mathbf{I}_s^n\mathbf{V}^{n-2})\right.\right. \\ &\qquad\left.\left.+d\tfrac{\mathbf{I}_s^n\mathbf{Q}^{n-3} - \mathbf{I}_s^n\mathbf{Q}^{n-1}}{\Delta t} \circ (\mathbf{I}_s^n\mathbf{V}^{n-3})\right] \right. \\ &\left.+[\mathbf{\Lambda}_s^n]^T\left[\mathbf{R}^n + \left((\mathbf{I}_s^n\mathbf{Q}^{n-1}) \circ \mathbf{C}_s^n + \beta\bar{\mathbf{C}}_s^n\right) \circ \mathbf{R}_{GCL}^n\right] + \left[\mathbf{\Lambda}_f^n\right]^T[\mathbf{A}^n\mathbf{Q}^n] + [\mathbf{\Lambda}_h^n]^T[\mathbf{P}^n\mathbf{Q}^n]\right\}\Delta t \\ &+\left(f^0 + \left[\mathbf{\Lambda}_g^0\right]^T\mathbf{G}^0 + \left[\mathbf{\Lambda}^0\right]^T\mathbf{R}^{in}\right)\Delta t.\end{aligned} \tag{53}$$

On time levels 1 and 2, the time derivatives are assumed to be discretized with the BDF1 and BDF2 schemes, respectively. Taking into account the dependencies on data at time levels $n-2$ and $n-3$, the adjoint equations are obtained as follows:

$$
\begin{aligned}
S:\ & \tfrac{a}{\Delta t}\mathbf{V}_s^n \circ \mathbf{C}_s^n \circ \mathbf{\Lambda}_s^n + \left[\tfrac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_s^n}\right]^T \mathbf{\Lambda}_s^n + [\mathbf{A}_s^n]^T \mathbf{\Lambda}_f^n + [\mathbf{P}_s^n]^T \mathbf{\Lambda}_h^n = \\
& - \left[\tfrac{\partial f}{\partial \mathbf{Q}_s^n}\right]^T - \mathbf{I}_s^n \Big\{ \left[\mathbf{I}_s^{n+1}\right]^T \left[\left(-\tfrac{a}{\Delta t}\mathbf{V}_s^{n+1} - \tfrac{c}{\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-1} - \tfrac{d}{\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-2} + \mathbf{R}_{GCL}^{n+1}\right) \circ \mathbf{C}_s^{n+1} \circ \mathbf{\Lambda}_s^{n+1}\right] \\
& + \left[\mathbf{I}_s^{n+2}\right]^T \left[\left(\tfrac{c}{\Delta t}\mathbf{I}_s^{n+2}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+2} \circ \mathbf{\Lambda}_s^{n+2}\right] + \left[\mathbf{I}_s^{n+3}\right]^T \left[\left(\tfrac{d}{\Delta t}\mathbf{I}_s^{n+3}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+3} \circ \mathbf{\Lambda}_s^{n+3}\right] \Big\}, \\
F:\ & \left[\tfrac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_f^n}\right]^T \mathbf{\Lambda}_s^n + \left[\mathbf{A}_f^n\right]^T \mathbf{\Lambda}_f^n + \left[\mathbf{P}_f^n\right]^T \mathbf{\Lambda}_h^n = \\
& - \left[\tfrac{\partial f}{\partial \mathbf{Q}_f^n}\right]^T - \mathbf{I}_f^n \Big\{ \left[\mathbf{I}_s^{n+1}\right]^T \left[\left(-\tfrac{a}{\Delta t}\mathbf{V}_s^{n+1} - \tfrac{c}{\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-1} - \tfrac{d}{\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-2} + \mathbf{R}_{GCL}^{n+1}\right) \circ \mathbf{C}_s^{n+1} \circ \mathbf{\Lambda}_s^{n+1}\right] \\
& + \left[\mathbf{I}_s^{n+2}\right]^T \left[\left(\tfrac{c}{\Delta t}\mathbf{I}_s^{n+2}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+2} \circ \mathbf{\Lambda}_s^{n+2}\right] + \left[\mathbf{I}_s^{n+3}\right]^T \left[\left(\tfrac{d}{\Delta t}\mathbf{I}_s^{n+3}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+3} \circ \mathbf{\Lambda}_s^{n+3}\right] \Big\}, \\
H:\ & \left[\tfrac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_h^n}\right]^T \mathbf{\Lambda}_s^n + [\mathbf{A}_h^n]^T \mathbf{\Lambda}_f^n + [\mathbf{P}_h^n]^T \mathbf{\Lambda}_h^n = \\
& - \left[\tfrac{\partial f}{\partial \mathbf{Q}_h^n}\right]^T - \mathbf{I}_h^n \Big\{ \left[\mathbf{I}_s^{n+1}\right]^T \left[\left(-\tfrac{a}{\Delta t}\mathbf{V}_s^{n+1} - \tfrac{c}{\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-1} - \tfrac{d}{\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-2} + \mathbf{R}_{GCL}^{n+1}\right) \circ \mathbf{C}_s^{n+1} \circ \mathbf{\Lambda}_s^{n+1}\right] \\
& + \left[\mathbf{I}_s^{n+2}\right]^T \left[\left(\tfrac{c}{\Delta t}\mathbf{I}_s^{n+2}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+2} \circ \mathbf{\Lambda}_s^{n+2}\right] + \left[\mathbf{I}_s^{n+3}\right]^T \left[\left(\tfrac{d}{\Delta t}\mathbf{I}_s^{n+3}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+3} \circ \mathbf{\Lambda}_s^{n+3}\right] \Big\},
\end{aligned}
\tag{54}
$$

$for\ 3 \le n \le N;$

$$
\begin{aligned}
S:\ & \tfrac{3}{2\Delta t}\mathbf{V}_s^n \circ \mathbf{C}_s^n \circ \mathbf{\Lambda}_s^n + \left[\tfrac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_s^n}\right]^T \mathbf{\Lambda}_s^n + [\mathbf{A}_s^n]^T \mathbf{\Lambda}_f^n + [\mathbf{P}_s^n]^T \mathbf{\Lambda}_h^n = \\
& - \left[\tfrac{\partial f}{\partial \mathbf{Q}_s^n}\right]^T - \mathbf{I}_s^n \Big\{ \left[\mathbf{I}_s^{n+1}\right]^T \left[\left(-\tfrac{a}{\Delta t}\mathbf{V}_s^{n+1} - \tfrac{c}{\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-1} - \tfrac{d}{\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-2} + \mathbf{R}_{GCL}^{n+1}\right) \circ \mathbf{C}_s^{n+1} \circ \mathbf{\Lambda}_s^{n+1}\right] \\
& + \left[\mathbf{I}_s^{n+2}\right]^T \left[\left(\tfrac{c}{\Delta t}\mathbf{I}_s^{n+2}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+2} \circ \mathbf{\Lambda}_s^{n+2}\right] + \left[\mathbf{I}_s^{n+3}\right]^T \left[\left(\tfrac{d}{\Delta t}\mathbf{I}_s^{n+3}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+3} \circ \mathbf{\Lambda}_s^{n+3}\right] \Big\}, \\
F:\ & \left[\tfrac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_f^n}\right]^T \mathbf{\Lambda}_s^n + \left[\mathbf{A}_f^n\right]^T \mathbf{\Lambda}_f^n + \left[\mathbf{P}_f^n\right]^T \mathbf{\Lambda}_h^n = \\
& - \left[\tfrac{\partial f}{\partial \mathbf{Q}_f^n}\right]^T - \mathbf{I}_f^n \Big\{ \left[\mathbf{I}_s^{n+1}\right]^T \left[\left(-\tfrac{a}{\Delta t}\mathbf{V}_s^{n+1} - \tfrac{c}{\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-1} - \tfrac{d}{\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-2} + \mathbf{R}_{GCL}^{n+1}\right) \circ \mathbf{C}_s^{n+1} \circ \mathbf{\Lambda}_s^{n+1}\right] \\
& + \left[\mathbf{I}_s^{n+2}\right]^T \left[\left(\tfrac{c}{\Delta t}\mathbf{I}_s^{n+2}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+2} \circ \mathbf{\Lambda}_s^{n+2}\right] + \left[\mathbf{I}_s^{n+3}\right]^T \left[\left(\tfrac{d}{\Delta t}\mathbf{I}_s^{n+3}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+3} \circ \mathbf{\Lambda}_s^{n+3}\right] \Big\}, \\
H:\ & \left[\tfrac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_h^n}\right]^T \mathbf{\Lambda}_s^n + [\mathbf{A}_h^n]^T \mathbf{\Lambda}_f^n + [\mathbf{P}_h^n]^T \mathbf{\Lambda}_h^n = \\
& - \left[\tfrac{\partial f}{\partial \mathbf{Q}_h^n}\right]^T - \mathbf{I}_h^n \Big\{ \left[\mathbf{I}_s^{n+1}\right]^T \left[\left(-\tfrac{a}{\Delta t}\mathbf{V}_s^{n+1} - \tfrac{c}{\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-1} - \tfrac{d}{\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-2} + \mathbf{R}_{GCL}^{n+1}\right) \circ \mathbf{C}_s^{n+1} \circ \mathbf{\Lambda}_s^{n+1}\right] \\
& + \left[\mathbf{I}_s^{n+2}\right]^T \left[\left(\tfrac{c}{\Delta t}\mathbf{I}_s^{n+2}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+2} \circ \mathbf{\Lambda}_s^{n+2}\right] + \left[\mathbf{I}_s^{n+3}\right]^T \left[\left(\tfrac{d}{\Delta t}\mathbf{I}_s^{n+3}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+3} \circ \mathbf{\Lambda}_s^{n+3}\right] \Big\},
\end{aligned}
\tag{55}
$$

$for\ n = 2;$

$$
\begin{aligned}
S:\ & \tfrac{1}{\Delta t}\mathbf{V}_s^n \circ \mathbf{C}_s^n \circ \mathbf{\Lambda}_s^n + \left[\tfrac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_s^n}\right]^T \mathbf{\Lambda}_s^n + \left[\mathbf{A}_s^n\right]^T \mathbf{\Lambda}_f^n + \left[\mathbf{P}_s^n\right]^T \mathbf{\Lambda}_h^n = \\
& -\left[\tfrac{\partial f}{\partial \mathbf{Q}_s^n}\right]^T - \mathbf{I}_s^n \left\{ \left[\mathbf{I}_s^{n+1}\right]^T \left[\left(-\tfrac{3}{2\Delta t}\mathbf{V}_s^{n+1} - \tfrac{1}{2\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-1} + \mathbf{R}_{GCL}^{n+1}\right) \circ \mathbf{C}_s^{n+1} \circ \mathbf{\Lambda}_s^{n+1}\right] \right. \\
& \left. + \left[\mathbf{I}_s^{n+2}\right]^T \left[\left(\tfrac{c}{\Delta t}\mathbf{I}_s^{n+2}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+2} \circ \mathbf{\Lambda}_s^{n+2}\right] + \left[\mathbf{I}_s^{n+3}\right]^T \left[\left(\tfrac{d}{\Delta t}\mathbf{I}_s^{n+3}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+3} \circ \mathbf{\Lambda}_s^{n+3}\right] \right\}, \\
\\
F:\ & \left[\tfrac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_f^n}\right]^T \mathbf{\Lambda}_s^n + \left[\mathbf{A}_f^n\right]^T \mathbf{\Lambda}_f^n + \left[\mathbf{P}_f^n\right]^T \mathbf{\Lambda}_h^n = \\
& -\left[\tfrac{\partial f}{\partial \mathbf{Q}_f^n}\right]^T - \mathbf{I}_f^n \left\{ \left[\mathbf{I}_s^{n+1}\right]^T \left[\left(-\tfrac{3}{2\Delta t}\mathbf{V}_s^{n+1} - \tfrac{1}{2\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-1} + \mathbf{R}_{GCL}^{n+1}\right) \circ \mathbf{C}_s^{n+1} \circ \mathbf{\Lambda}_s^{n+1}\right] \right. \\
& \left. + \left[\mathbf{I}_s^{n+2}\right]^T \left[\left(\tfrac{c}{\Delta t}\mathbf{I}_s^{n+2}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+2} \circ \mathbf{\Lambda}_s^{n+2}\right] + \left[\mathbf{I}_s^{n+3}\right]^T \left[\left(\tfrac{d}{\Delta t}\mathbf{I}_s^{n+3}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+3} \circ \mathbf{\Lambda}_s^{n+3}\right] \right\}, \\
\\
H:\ & \left[\tfrac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_h^n}\right]^T \mathbf{\Lambda}_s^n + \left[\mathbf{A}_h^n\right]^T \mathbf{\Lambda}_f^n + \left[\mathbf{P}_h^n\right]^T \mathbf{\Lambda}_h^n = \\
& -\left[\tfrac{\partial f}{\partial \mathbf{Q}_h^n}\right]^T - \mathbf{I}_h^n \left\{ \left[\mathbf{I}_s^{n+1}\right]^T \left[\left(-\tfrac{3}{2\Delta t}\mathbf{V}_s^{n+1} - \tfrac{1}{2\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-1} + \mathbf{R}_{GCL}^{n+1}\right) \circ \mathbf{C}_s^{n+1} \circ \mathbf{\Lambda}_s^{n+1}\right] \right. \\
& \left. + \left[\mathbf{I}_s^{n+2}\right]^T \left[\left(\tfrac{c}{\Delta t}\mathbf{I}_s^{n+2}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+2} \circ \mathbf{\Lambda}_s^{n+2}\right] + \left[\mathbf{I}_s^{n+3}\right]^T \left[\left(\tfrac{d}{\Delta t}\mathbf{I}_s^{n+3}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+3} \circ \mathbf{\Lambda}_s^{n+3}\right] \right\}, \\
& \qquad for\ n = 1;
\end{aligned}
\tag{56}
$$

$$
\begin{aligned}
& \left[\tfrac{\partial \mathbf{R}^{in}}{\partial \mathbf{Q}^0}\right]^T \mathbf{\Lambda}^0 = -\left[\tfrac{\partial f}{\partial \mathbf{Q}^0}\right]^T - \left\{ \left[\mathbf{I}_s^1\right]^T \left[\left(-\tfrac{1}{\Delta t}\mathbf{V}_s^1 + \mathbf{R}_{GCL}^1\right) \circ \mathbf{C}_s^1 \circ \mathbf{\Lambda}_s^1\right] \right. \\
& \left. + \left[\mathbf{I}_s^2\right]^T \left[\left(\tfrac{1}{2\Delta t}\mathbf{I}_s^2\mathbf{V}^0\right) \circ \mathbf{C}_s^2 \circ \mathbf{\Lambda}_s^2\right] + \left[\mathbf{I}_s^3\right]^T \left[\left(\tfrac{d}{\Delta t}\mathbf{I}_s^3\mathbf{V}^0\right) \circ \mathbf{C}_s^3 \circ \mathbf{\Lambda}_s^3\right] \right\}, \\
& \qquad for\ n = 0.
\end{aligned}
$$

The corresponding grid adjoint equations are obtained as follows. Assuming $\mathbf{\Lambda}^{N+1} = \mathbf{\Lambda}^{N+2} = \mathbf{\Lambda}^{N+3} = 0$:

$$
\begin{aligned}
&-\left[\tfrac{\partial \mathbf{G}^n}{\partial \mathbf{X}^n}\right]^T \mathbf{\Lambda}_g^n = \left[a\tfrac{\mathbf{Q}_s^n - \mathbf{I}_s^n \mathbf{Q}^{n-1}}{\Delta t} \odot \tfrac{\partial \mathbf{V}_s^n}{\partial \mathbf{X}^n}\right]^T \left(\mathbf{C}_s^n \circ \mathbf{\Lambda}_s^n\right) \\
&+\left[c\tfrac{\mathbf{I}_s^{n+2}\mathbf{Q}^n - \mathbf{I}_s^{n+2}\mathbf{Q}^{n+1}}{\Delta t} \odot \left(\mathbf{I}_s^{n+2}\tfrac{\partial \mathbf{V}^n}{\partial \mathbf{X}^n}\right)\right]^T \left(\mathbf{C}_s^{n+2} \circ \mathbf{\Lambda}_s^{n+2}\right) \\
&+\left[d\tfrac{\mathbf{I}_s^{n+3}\mathbf{Q}^n - \mathbf{I}_s^{n+3}\mathbf{Q}^{n+1}}{\Delta t} \odot \left(\mathbf{I}_s^{n+3}\tfrac{\partial \mathbf{V}^n}{\partial \mathbf{X}^n}\right)\right]^T \left(\mathbf{C}_s^{n+3} \circ \mathbf{\Lambda}_s^{n+3}\right) + \left[\tfrac{\partial (\mathbf{A}^n \mathbf{Q}^n)}{\partial \mathbf{X}^n}\right]^T \mathbf{\Lambda}_f^n \\
&+\sum_{k=0}^{3}\left[\tfrac{\partial \mathbf{R}^{n+k}}{\partial \mathbf{X}^n} + \left((\mathbf{I}_s^{n+k}\mathbf{Q}^{n+k-1}) \circ \mathbf{C}_s^{n+k} + \beta \bar{\mathbf{C}}_s^{n+k}\right) \odot \tfrac{\partial \mathbf{R}_{GCL}^{n+k}}{\partial \mathbf{X}^n}\right]^T \mathbf{\Lambda}_s^{n+k} + \left[\tfrac{\partial f}{\partial \mathbf{X}^n}\right]^T, \qquad for\ 3 \le n \le N;
\end{aligned}
$$

$$
\begin{aligned}
&-\left[\tfrac{\partial \mathbf{G}^n}{\partial \mathbf{X}^n}\right]^T \mathbf{\Lambda}_g^n = \left[\tfrac{3}{2}\tfrac{\mathbf{Q}_s^n - \mathbf{I}_s^n \mathbf{Q}^{n-1}}{\Delta t} \odot \tfrac{\partial \mathbf{V}_s^n}{\partial \mathbf{X}^n}\right]^T \left(\mathbf{C}_s^n \circ \mathbf{\Lambda}_s^n\right) \\
&+\left[c\tfrac{\mathbf{I}_s^{n+2}\mathbf{Q}^n - \mathbf{I}_s^{n+2}\mathbf{Q}^{n+1}}{\Delta t} \odot \left(\mathbf{I}_s^{n+2}\tfrac{\partial \mathbf{V}^n}{\partial \mathbf{X}^n}\right)\right]^T \left(\mathbf{C}_s^{n+2} \circ \mathbf{\Lambda}_s^{n+2}\right) \\
&+\left[d\tfrac{\mathbf{I}_s^{n+3}\mathbf{Q}^n - \mathbf{I}_s^{n+3}\mathbf{Q}^{n+1}}{\Delta t} \odot \left(\mathbf{I}_s^{n+3}\tfrac{\partial \mathbf{V}^n}{\partial \mathbf{X}^n}\right)\right]^T \left(\mathbf{C}_s^{n+3} \circ \mathbf{\Lambda}_s^{n+3}\right) + \left[\tfrac{\partial (\mathbf{A}^n \mathbf{Q}^n)}{\partial \mathbf{X}^n}\right]^T \mathbf{\Lambda}_f^n \\
&+\sum_{k=0}^{3}\left[\tfrac{\partial \mathbf{R}^{n+k}}{\partial \mathbf{X}^n} + \left((\mathbf{I}_s^{n+k}\mathbf{Q}^{n+k-1}) \circ \mathbf{C}_s^{n+k} + \beta \bar{\mathbf{C}}_s^{n+k}\right) \odot \tfrac{\partial \mathbf{R}_{GCL}^{n+k}}{\partial \mathbf{X}^n}\right]^T \mathbf{\Lambda}_s^{n+k} + \left[\tfrac{\partial f}{\partial \mathbf{X}^n}\right]^T, \qquad for\ n = 2;
\end{aligned}
$$

$$
\begin{aligned}
&-\left[\tfrac{\partial \mathbf{G}^n}{\partial \mathbf{X}^n}\right]^T \mathbf{\Lambda}_g^n = \left[\tfrac{\mathbf{Q}_s^n - \mathbf{I}_s^n \mathbf{Q}^{n-1}}{\Delta t} \odot \tfrac{\partial \mathbf{V}_s^n}{\partial \mathbf{X}^n}\right]^T \left(\mathbf{C}_s^n \circ \mathbf{\Lambda}_s^n\right) \\
&+\left[c\tfrac{\mathbf{I}_s^{n+2}\mathbf{Q}^n - \mathbf{I}_s^{n+2}\mathbf{Q}^{n+1}}{\Delta t} \odot \left(\mathbf{I}_s^{n+2}\tfrac{\partial \mathbf{V}^n}{\partial \mathbf{X}^n}\right)\right]^T \left(\mathbf{C}_s^{n+2} \circ \mathbf{\Lambda}_s^{n+2}\right) \\
&+\left[d\tfrac{\mathbf{I}_s^{n+3}\mathbf{Q}^n - \mathbf{I}_s^{n+3}\mathbf{Q}^{n+1}}{\Delta t} \odot \left(\mathbf{I}_s^{n+3}\tfrac{\partial \mathbf{V}^n}{\partial \mathbf{X}^n}\right)\right]^T \left(\mathbf{C}_s^{n+3} \circ \mathbf{\Lambda}_s^{n+3}\right) + \left[\tfrac{\partial (\mathbf{A}^n \mathbf{Q}^n)}{\partial \mathbf{X}^n}\right]^T \mathbf{\Lambda}_f^n \\
&+\sum_{k=0}^{3}\left[\tfrac{\partial \mathbf{R}^{n+k}}{\partial \mathbf{X}^n} + \left((\mathbf{I}_s^{n+k}\mathbf{Q}^{n+k-1}) \circ \mathbf{C}_s^{n+k} + \beta \bar{\mathbf{C}}_s^{n+k}\right) \odot \tfrac{\partial \mathbf{R}_{GCL}^{n+k}}{\partial \mathbf{X}^n}\right]^T \mathbf{\Lambda}_s^{n+k} + \left[\tfrac{\partial f}{\partial \mathbf{X}^n}\right]^T, \qquad for\ n = 1;
\end{aligned}
$$

$$
\begin{aligned}
&-\left[\tfrac{\partial \mathbf{G}^0}{\partial \mathbf{X}^0}\right]^T \mathbf{\Lambda}_g^0 = \sum_{n=1}^{N}\left[\tfrac{\partial \mathbf{G}^n}{\partial \mathbf{X}^0}\right]^T \mathbf{\Lambda}_g^n + \left[\tfrac{1}{2}\tfrac{\mathbf{I}_s^2\mathbf{Q}^0 - \mathbf{I}_s^2\mathbf{Q}^1}{\Delta t} \odot \left(\mathbf{I}_s^2\tfrac{\partial \mathbf{V}^0}{\partial \mathbf{X}^0}\right)\right]^T \left(\mathbf{C}_s^{n+2} \circ \mathbf{\Lambda}_s^{n+2}\right) \\
&+\left[d\tfrac{\mathbf{I}_s^3\mathbf{Q}^0 - \mathbf{I}_s^3\mathbf{Q}^1}{\Delta t} \odot \left(\mathbf{I}_s^3\tfrac{\partial \mathbf{V}^0}{\partial \mathbf{X}^0}\right)\right]^T \left(\mathbf{C}_s^{n+3} \circ \mathbf{\Lambda}_s^3\right) + \left[\tfrac{\partial R^{in}}{\partial \mathbf{X}^0}\right]^T \mathbf{\Lambda}^0 \\
&+\sum_{k=1}^{3}\left[\tfrac{\partial \mathbf{R}^k}{\partial \mathbf{X}^0} + \left((\mathbf{I}_s^k\mathbf{Q}^{k-1}) \circ \mathbf{C}_s^k + \beta \bar{\mathbf{C}}_s^k\right) \odot \tfrac{\partial \mathbf{R}_{GCL}^k}{\partial \mathbf{X}^0}\right]^T \mathbf{\Lambda}_s^k + \left[\tfrac{\partial f}{\partial \mathbf{X}^0}\right]^T, \qquad for\ n = 0.
\end{aligned}
\tag{57}
$$

The sensitivity derivative for the higher-order BDF schemes is evaluated using Eq. 36.

References

[1]Newman III, J., Taylor III, A., Barnwell, R., Newman, P., and Hou, G.-W., "Overview of Sensitivity Analysis and Shape Optimization for Complex Aerodynamic Configurations," *AIAA Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 87–96.

[2]Peter, J. and Dwight, R., "Numerical Sensitivity Analysis for Aerodynamic Optimization: A Survey of Approaches," *Computers and Fluids*, Vol. 39, No. 3, 2010, pp. 373–391.

[3]Nielsen, E. and Jones, W., "Integrated Design of an Active Flow Control System Using a Time-Dependent Adjoint Method," *Mathematical Modeling of Natural Phenomena*, Vol. 6, No. 3, 2011, pp. 1195–1206.

[4]Rumpfkeil, M. and Zingg, D., "General Framework for the Optimal Control of Unsteady Flows with Applications," AIAA Paper 2007–1128, 2007.

[5]Nielsen, E., Diskin, B., and Yamaleev, N., "Discrete Adjoint-Based Design Optimization of Unsteady Turbulent Flows on Dynamic Unstructured Grids," *AIAA Journal*, Vol. 48, No. 6, 2010, pp. 1195–1206.

[6]Nadarajah, S. and Jameson, A., "Optimal Control of Unsteady Flows Using Time Accurate and Non-Linear Frequency Domain Methods," AIAA Paper 2002–5436, 2002.

[7]Muldoon, F., "Control of a Simplified Unsteady Film-Cooling Flow Using Gradient-Based Optimization," *AIAA Journal*, Vol. 46, No. 10, 2008, pp. 2443–2458.

[8]Mavriplis, D., "Solution of the Unsteady Discrete Adjoint for Three-Dimensional Problems on Dynamically Deforming Unstructured Meshes," AIAA Paper 2008–0727, 2008.

[9]Choi, S., Potsdam, M., Lee, K., Iaccarino, G., and Alonso, J., "Helicopter Rotor Design Using a Time-Spectral and Adjoint-Based Method," AIAA Paper 2008–5810, 2008.

[10]Hicks, R. and Henne, P., "Wing Design by Numerical Optimization," *AIAA Journal of Aircraft*, Vol. 15, No. 7, 1978, pp. 407–412.

[11]Baysal, O. and Eleshaky, M., "Aerodynamic Sensitivity Analysis Methods for the Compressible Euler Equations," *ASME Journal of Fluids Engineering*, Vol. 113, No. 4, 1991, pp. 681–688.

[12]Lyness, J., "Numerical Algorithms Based on the Theory of Complex Variables," *Proceedings of the 22nd ACM National Conference*, 1967.

[13]Pironneau, O., "On Optimum Profiles in Stokes Flow," *Journal of Fluid Mechanics*, Vol. 59, No. 1, 1973, pp. 117–128.

[14]Fidkowski, K. J. and Darmofal, D. L., "Review of Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics," *AIAA Journal*, Vol. 49, No. 4, 2011, pp. 673–694.

[15]Wang, Q., *Uncertainty Quantification for Unsteady Fluid Flow Using Adjoint-Based Approaches*, Ph.D. thesis, Stanford University, Dec. 2008.

[16]Rallabhandi, S., "Sonic Boom Adjoint Methodology and Its Applications," AIAA Paper 2011–3497, 2011.

[17]Lee, J.-D. and Jameson, A., "Natural-Laminar-Flow Airfoil and Wing Design by Adjoint Method and Automatic Transition Prediction," AIAA Paper 2009–0897, 2009.

[18]Nielsen, E. and Anderson, W., "Aerodynamic Design Optimization On Unstructured Meshes Using the Navier-Stokes Equations," *AIAA Journal*, Vol. 37, No. 11, 1999, pp. 1411–1419.

[19]Nielsen, E. J. and Anderson, W. K., "Recent Improvements in Aerodynamic Design Optimization on Unstructured Meshes," *AIAA Journal*, Vol. 40, No. 6, 2002, pp. 1155–1163.

[20]Nielsen, E., Lee-Rausch, E., and Jones, W., "Adjoint-Based Design of Rotors Using the Navier-Stokes Equations in a Noninertial Reference Frame," *AIAA Journal of Aircraft*, Vol. 47, No. 2, 2010, pp. 638–646.

[21]Nielsen, E. J. and Kleb, W. L., "Efficient Construction of Discrete Adjoint Operators on Unstructured Grids Using Complex Variables," *AIAA Journal*, Vol. 44, No. 4, 2006, pp. 827–836.

[22]Nielsen, E. J. and Park, M. A., "Using An Adjoint Approach to Eliminate Mesh Sensitivities in Computational Design," *AIAA Journal*, Vol. 44, No. 5, 2006, pp. 948–953.

[23]Nielsen, E. J., *Aerodynamic Design Sensitivities on an Unstructured Mesh Using the Navier-Stokes Equations and a Discrete Adjoint Formulation*, Ph.D. thesis, Virginia Polytechnic Institute and State University, Dec. 1998.

[24]Nielsen, E. J., Lu, J., Park, M. A., and Darmofal, D. L., "An Implicit, Exact Dual Adjoint Solution Method for Turbulent Flows on Unstructured Grids," *Computers and Fluids*, Vol. 33, No. 9, 2004, pp. 1131–1155.

[25]Lee, B. and Kim, C., "Aerodynamic Shape Optimization Using Discrete Adjoint Formulation Based on Overset Mesh Technique," *European Conference on Computational Fluid Dynamics ECCOMAS CFD 2006*, Sept. 2006.

[26]Elliott, J., "Discrete Adjoint Analysis and Optimization with Overset Grid Modeling of the Compressible High-Re Navier-Stokes Equations," *Proceedings of the 6th Overset Grid and Solution Technology Symposium*, Oct. 2002.

[27]Lei, Z. and Makino, Y., "CFD-Based Aerodynamic Shape Optimization for Japanese Experimental Supersonic Transport," ICAS 2004–4.3.3, Sept. 2004.

[28]Liao, W. and Tsai, H., "Aerodynamic Design Optimization by the Adjoint Equation Method on Overset Grids," AIAA Paper 2006–0054, 2006.

[29]Cambier, L. and Veuillot, J.-P., "Status of the elsA CFD Software for Flow Simulation and Multidisciplinary Applications," AIAA Paper 2008–0664, 2008.

[30]O'Brien Jr., D. M., *Analysis of Computational Modeling Techniques for Complete Rotorcraft Configurations*, Ph.D. thesis, Georgia Institute of Technology, May 2006.

[31]Horn, R. A. and Johnson, C. R., *Matrix Analysis*, Cambridge University Press, New York, NY, 1985.

[32]Chorin, A., "A Numerical Method for Solving Incompressible Viscous Flow Problems," *Journal of Computational Physics*, Vol. 2, 1967, pp. 12–26.

[33]Nyukhtikov, M., Smelova, N., Mitchell, B., and Holmes, D., "Optimized Dual-Time Stepping Technique for Time-Accurate Navier-Stokes Calculation," *Proceedings of the 10th International Symposium on Unstructured Aerodynamics, Aeroacoustics, and Aeroelastics of Turbomachinery*, 2003.

[34]Vatsa, V., Carpenter, M., and Lockard, D., "Re-evaluation of an Optimized Second Order Backward Difference (BDF2OPT) Scheme for Unsteady Flow Applications," AIAA Paper 2010–0122, 2010.

[35]Hirt, C., Amsden, A., and Cook, J., "An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds," *Journal of Computational Physics*, Vol. 14, No. 3, 1974, pp. 227–257.

[36]Biedron, R. and Thomas, J., "Recent Enhancements to the FUN3D Flow Solver for Moving-Mesh Applications," AIAA Paper 2009–1360, 2009.

[37]Thomas, P. and Lombard, C., "Geometrical Conservation Law and Its Application to Flow Computations on Moving Grids," *AIAA Journal*, Vol. 17, No. 10, 1979, pp. 1030–1037.

[38]Noack, R., Boger, D., Kunz, R., and Carrica, P., "SUGGAR++: An Improved General Overset Grid Assembly Capability," AIAA Paper 2009–3992, 2009.

[39]Biedron, R. and Lee-Rausch, E., "Computation of UH-60A Airloads Using CFD/CSD Coupling On Unstructured Meshes," *American Helicopter Society 67th Annual Forum Proceedings*, 2011.

[40]Lee-Rausch, E. and Biedron, R., "Simulation of an Isolated Tiltrotor in Hover with an Unstructured Overset-Grid RANS Solver," *American Helicopter Society 65th Annual Forum Proceedings*, 2009.

[41]Biedron, R. and Lee-Rausch, E., "Rotor Airloads Prediction Using Unstructured Meshes and Loose CFD/CSD Coupling," AIAA Paper 2008–7341, 2008.

[42]O'Brien Jr., D. and Smith, M., "Analysis of Rotor-Fuselage Interactions Using Various Rotor Models," AIAA Paper 2005–0468, 2005.

[43]Lynch, E., *Advanced CFD Methods for Wind Turbine Analysis*, Ph.D. thesis, Georgia Institute of Technology, Jan. 2011.

[44]Abras, J., *Enhancement of Aeroelastic Rotor Airload Prediction Methods*, Ph.D. thesis, Georgia Institute of Technology, May 2009.

[45]Abras, J., Lynch, C., and Smith, M., "Advances in Rotorcraft Simulations with Unstructured CFD," *American Helicopter Society 63rd Annual Forum Proceedings*, 2007.

[46]Noack, R., "DiRTlib: A Library to Add an Overset Capability to Your Flow Solver," AIAA Paper 2005–5116, 2005.

[47]Anderson, W. K. and Bonhaus, D. L., "An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids," *Computers and Fluids*, Vol. 23, No. 1, 1994, pp. 1–22.

[48]http://fun3d.larc.nasa.gov, Last accessed December 20, 2011.

[49]Anderson, W., Rausch, R., and Bonhaus, D., "Implicit/Multigrid Algorithms for Incompressible Turbulent Flows on Unstructured Grids," *Journal of Computational Physics*, Vol. 128, No. 2, 1996, pp. 391–408.

[50]Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, 1981, pp. 357–372.

[51]Neel, R., Godfrey, A., and McGrory, W., "Low-Speed, Time-Accurate Validation of GASP Version 4," AIAA Paper 2005–686, 2005.

[52]Spalart, P. R. and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," *La Recherche Aerospatiale*, Vol. 1, No. 1, 1994, pp. 5–21.

[53]Hammond, D., Private communication, 2011.

[54]Saad, Y. and Schultz, M. H., "GMRES: A Generalized Minimum Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal of Scientific and Statistical Computing*, Vol. 7, 1986, pp. 856–869.

[55]Samareh, J. A., *A Novel Shape Parameterization Approach*, NASA/TM-1999-209116, May 1999.

[56]Yamaleev, N., Diskin, B., and Nielsen, E., "Adjoint-Based Methodology for Time-Dependent Optimization," AIAA Paper 2008–5857, 2008.

[57]Lyness, J. and Moler, C., "Numerical Differentiation of Analytic Functions," *SIAM Journal of Numerical Analysis*, Vol. 4, 1967, pp. 202–210.

[58]Anderson, W., Newman, J., Whitfield, D., and Nielsen, E., "Sensitivity Analysis for the Navier-Stokes Equations on Unstructured Meshes Using Complex Variables," *AIAA Journal*, Vol. 39, No. 1, 2001, pp. 56–63.

[59]Kleb, W. L., Nielsen, E. J., Gnoffo, P. A., Park, M. A., and Wood, W. A., "Collaborative Software Development in Support of Fast Adaptive AeroSpace Tools (FAAST)," AIAA Paper 2003–3978, 2003.

[60]http://wiki.lustre.org/index.php/Main_Page, Last accessed December 20, 2011.

[61]Hand, M., Simms, D., Fingersh, L., Jager, D., Cotrell, J., Schreck, S., and Larwood, S., "Unsteady Aerodynamics Experiment Phase VI: Wind Tunnel Test Configurations and Available Data Campaigns," NREL/TP-500-29955, 2001.

[62]Jeong, J. and Hussain, F., "On the Identification of a Vortex," *Journal of Fluid Mechanics*, Vol. 285, 1985, pp. 69–94.

[63]Gill, P., Murray, W., Saunders, M., and Wright, M., "User's Guide for NPSOL 5.0: A FORTRAN Package for Nonlinear Programming," Technical Report SOL 94, 1995.

[64]Willmott, A. and Ellington, C., "The Mechanics of Flight in the Hawkmoth *Manduca Sexta* I. Kinematics of Hovering and Forward Flight," *Journal of Experimental Biology*, Vol. 200, 1997, pp. 2705–2722.

[65]McClung, A., *Influence of Structural Flexibility on Flapping Wing Propulsion*, Ph.D. thesis, Air Force Institute of Technology, Nov. 2009.

[66]Marcum, D. L. and Weatherill, N. P., "Unstructured Grid Generation Using Iterative Point Insertion and Local Reconnection," *AIAA Journal*, Vol. 33, No. 9, 1995, pp. 1619–1625.

[67]Kaufman, L. and Gay, D., "PORT Library: Optimization and Mathematical Programming – User's Manual," Bell Laboratories, 1997.

[68]Bousman, W. and Kufeld, R., "UH-60A Airloads Catalog," NASA TM 2005-212827, Aug. 2005.

[69]Venditti, D. A. and Darmofal, D. L., "Anisotropic Grid Adaptation for Functional Outputs: Application to Two-Dimensional Viscous Flows," *Journal of Computational Physics*, Vol. 187, 2003, pp. 22–46.

[70]Park, M. A., "Three–Dimensional Turbulent RANS Adjoint–Based Error Correction," AIAA Paper 2003–3849, 2003.

[71]Lee-Rausch, E. M., Park, M. A., Jones, W. T., Hammond, D. P., and Nielsen, E. J., "Application of a Parallel Adjoint-Based Error Estimation and Anisotropic Grid Adaptation for Three-Dimensional Aerospace Configurations," AIAA Paper 2005–4842, 2005.

[72]Park, M., Lee-Rausch, E., and Rumsey, C., "FUN3D and CFL3D Computations for the First High Lift Prediction Workshop," AIAA Paper 2011–0936, 2011.

[73]Yamaleev, N., Diskin, B., and Nielsen, E., "Local-in-Time Adjoint-Based Method for Design Optimization of Unsteady Flows," *Journal of Computational Physics*, Vol. 229, No. 14, 2010, pp. 5394–5407.

[74]Pathak, K. and Yamaleev, N., "POD-Based Reduced-Order Model for Arbitrary Mach Number Flows," AIAA Paper 2011–3111, 2011.

[75]Alexandrov, N., Dennis Jr., J., Lewis, R., and Torczon, V., "A Trust Region Framework for Managing the Use of Approximation Models in Optimization," *Structural Optimization*, Vol. 15, No. 1, 1998, pp. 16–23.

[76]Nishikawa, H. and Diskin, B., "Development and Application of Parallel Agglomerated Multigrid Methods for Complex Geometries," AIAA Paper 2011–3232, 2011.

Table 1. Coefficients for BDF schemes.

Scheme	a	b	c	d
BDF1	1	-1	0	0
BDF2	3/2	-2	1/2	0
BDF3	11/6	-3	3/2	$-1/3$
BDF2opt	1.66	-2.48	0.98	-0.16

Table 2. Values of the sensitivity derivative $\partial C_{L_5}/\partial$D for different design variables and temporal discretizations for compressible flow. The symbols A and C denote adjoint and complex-variable results, respectively. Discrepancies are shown in bold and underlined.

Variable	BDF1	BDF2	BDF2opt	BDF3
Angle of	A: 0.116458961683733	A: 0.102099965021956	A: 0.102915752531413	A: 0.103785048456802
Attack	C: 0.116458961683734	C: 0.102099965021956	C: 0.102915752531413	C: 0.103785048456802
Rot Rate	A: 0.619149219**921508**	A: 0.609270815**829788**	A: 0.592456231**940897**	A: 0.575091540**944799**
Blade 1	C: 0.619149219**933539**	C: 0.609270815**842755**	C: 0.592456231**953869**	C: 0.575091540**957581**
Shape	A: 0.056440771725**301**	A: 0.064382783171**893**	A: 0.062734653842**921**	A: 0.060943525618**014**
Blade 2	C: 0.056440771725**196**	C: 0.064382783171**802**	C: 0.062734653842**842**	C: 0.060943525617**920**
Flap Freq	A: -0.414712919056**299**	A: -0.337250987004**676**	A: -0.344555513267**488**	A: -0.352419586848**976**
Blade 3	C: -0.414712919056**270**	C: -0.337250987004**642**	C: -0.344555513267**474**	C: -0.352419586848**961**
Rot Rate	A: 6.86680217888**885**	A: 7.42798143738**984**	A: 7.31688305983**601**	A: 7.20812218587**293**
Fuselage	C: 6.86680217888**239**	C: 7.42798143738**254**	C: 7.31688305982**953**	C: 7.20812218586**623**
Trans Rate	A: 0.420300051382122	A: 0.400837175635065	A: 0.390973864106570	A: 0.379952931745697
Fuselage	C: 0.420300051369376	C: 0.400837175622066	C: 0.390973864093789	C: 0.379952931733500
Shape	A: -0.007809447236**753**	A: -0.009590444345**683**	A: -0.009613538492**229**	A: -0.009705401931**920**
Fuselage	C: -0.007809447236**691**	C: -0.009590444345**727**	C: -0.009613538492**351**	C: -0.009705401931**704**

Table 3. Values of the sensitivity derivative $\partial C_{L_5}/\partial \mathbf{D}$ for different design variables and temporal discretizations for incompressible flow. The symbols A and C denote adjoint and complex-variable results, respectively. Discrepancies are shown in bold and underlined.

Variable	BDF1	BDF2	BDF2opt	BDF3
Angle of	A: 0.000195945789030	A: 0.000234143173131	A: 0.000218182269639	A: 0.000191641169710
Attack	C: 0.000195945789030	C: 0.000234143173131	C: 0.000218182269639	C: 0.000191641169711
Rot Rate	A: 0.009518073976865	A: 0.010325090376673	A: 0.010544987182945	A: 0.010757597020150
Blade 1	C: 0.009518073976838	C: 0.010325090376647	C: 0.010544987182921	C: 0.010757597020128
Shape	A: 0.000535025241509	A: 0.000607314158464	A: 0.000618811948355	A: 0.000633736751875
Blade 2	C: 0.000535025241508	C: 0.000607314158463	C: 0.000618811948355	C: 0.000633736751875
Flap Freq	A: -0.004866399384562	A: -0.004825188859067	A: -0.004821787992149	A: -0.004810632891273
Blade 3	C: -0.004866399384562	C: -0.004825188859067	C: -0.004821787992149	C: -0.004810632891273
Rot Rate	A: 0.042649260159755	A: 0.044962632318017	A: 0.044947751807594	A: 0.044876653248215
Fuselage	C: 0.042649260159807	C: 0.044962632318090	C: 0.044947751807680	C: 0.044876653248312
Trans Rate	A: 0.010034159304733	A: 0.010404514410124	A: 0.010284602229241	A: 0.010043806857134
Fuselage	C: 0.010034159304771	C: 0.010404514410192	C: 0.010284602229293	C: 0.010043806857193
Shape	A: 0.000087061995334	A: 0.000079589134812	A: 0.000082271937020	A: 0.000086753178814
Fuselage	C: 0.000087061995336	C: 0.000079589134815	C: 0.000082271937019	C: 0.000086753178823

Table 4. Values of the initial and final design variables for the flapping wing configuration. The abbreviation COR denotes the center of rotation.

Variable	Baseline	Distribution Target Function	Time-Average Target Function
x-COR	0.000	$0.025c$	$0.027c$
y-COR	0.000	$-0.119c$	$-0.114c$
z-COR	0.000	$0.011c$	$0.012c$
A_x	0.00	0.77	-0.11
B_x	45.00	45.13	45.25
ω_{1x}	163.36	163.45	163.36
ω_{2x}	163.36	177.47	192.77
A_y	0.000	0.30	-0.99
B_y	0.000	-1.50	-0.26
ω_{1y}	163.36	162.76	163.15
ω_{2y}	163.36	163.10	162.97
A_z	-60.00	-62.71	-62.83
B_z	0.00	0.69	-1.55
ω_{1z}	163.36	173.59	189.57
ω_{2z}	163.36	164.41	163.55

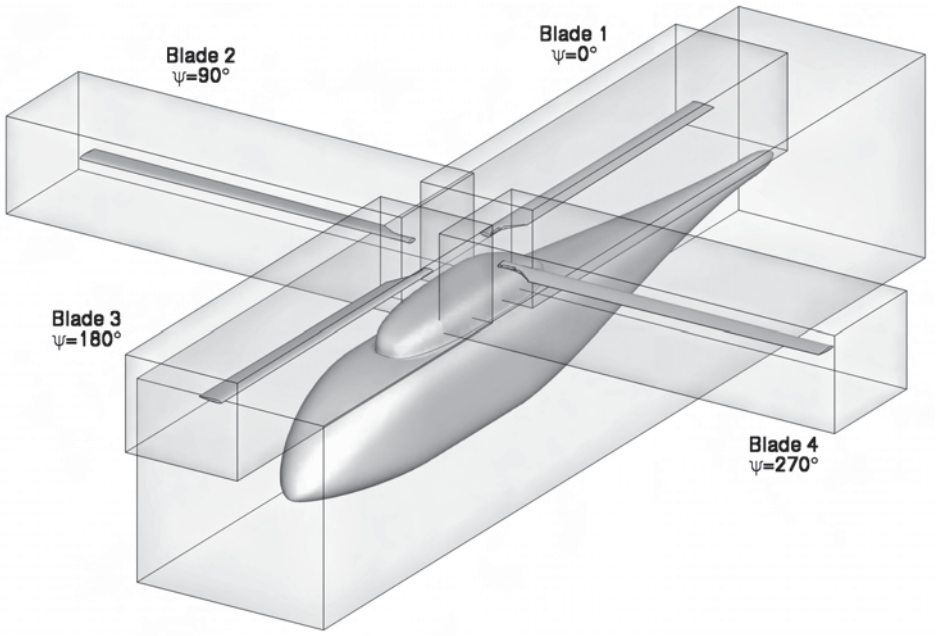


Figure 1. Nearfield view of geometry and composite grid system used for linearization accuracy study.

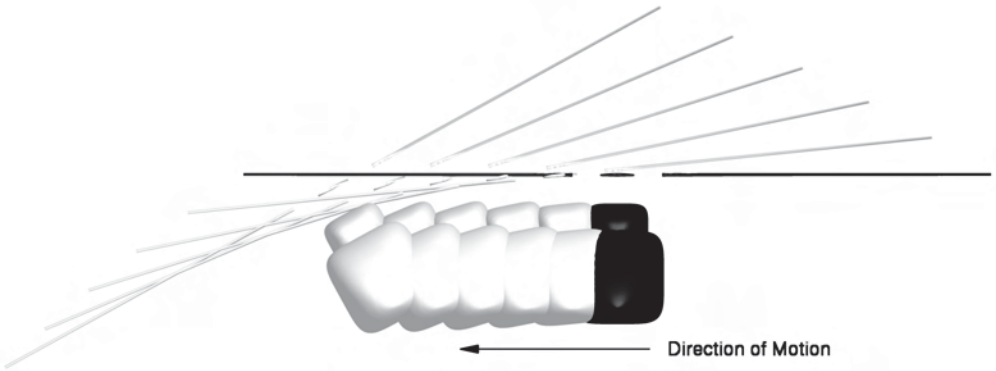


Figure 2. Imposed motion for linearization accuracy study. Geometry shown every 720 deg of rotor azimuth.

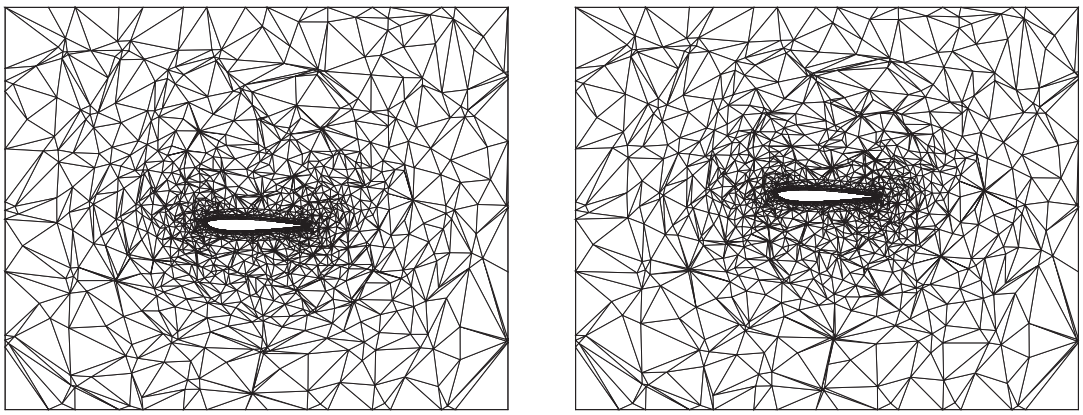


Figure 3. Cross-sections of deforming blade mesh showing maximum vertical displacements at blade tip during linearization accuracy study.

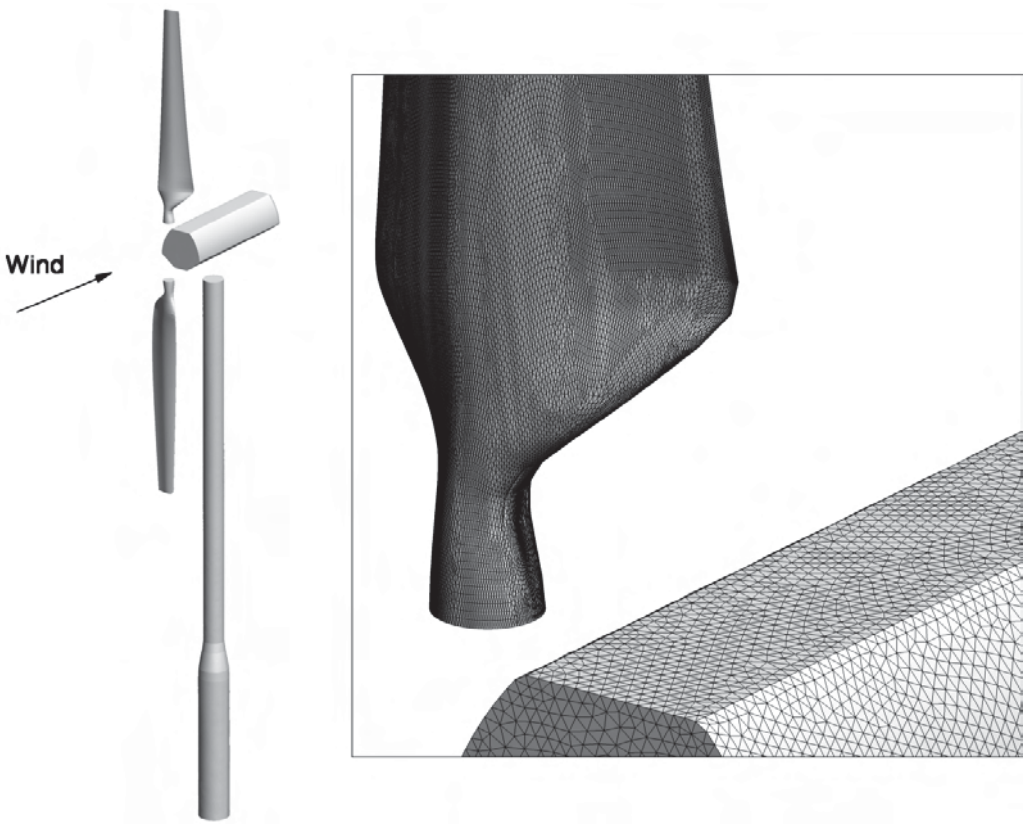


Figure 4. Wind turbine configuration and nearfield view of surface mesh in hub region.

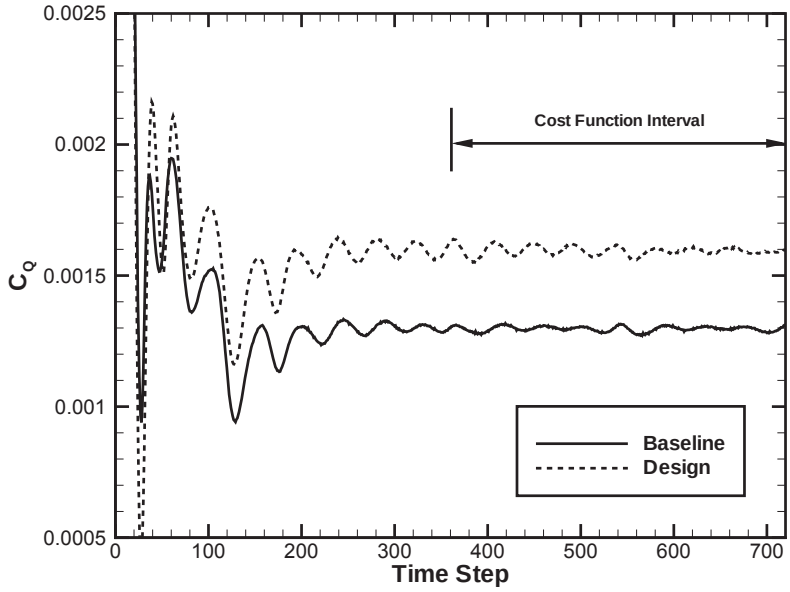


Figure 5. Baseline and final torque profiles for wind turbine configuration.

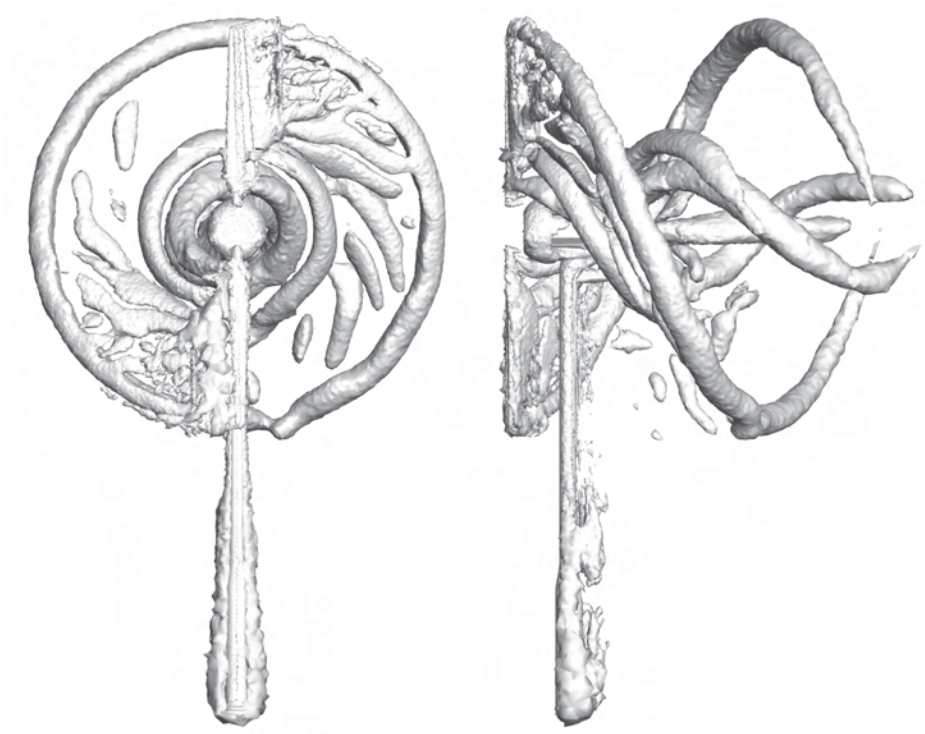


Figure 6. Front and side views of an isosurface of the Q criterion for the baseline wind turbine configuration.

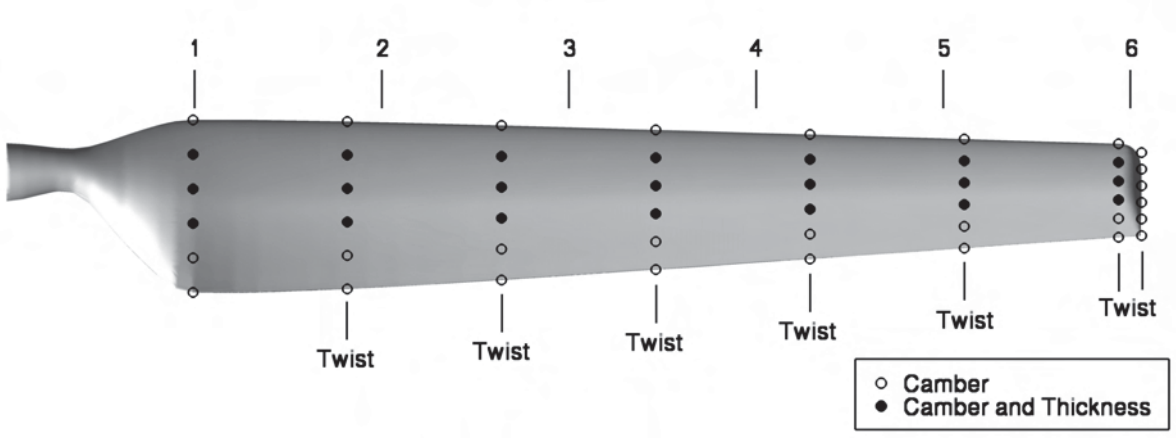


Figure 7. Blade planform geometry, shape variable locations, and spanwise stations for wind turbine configuration.

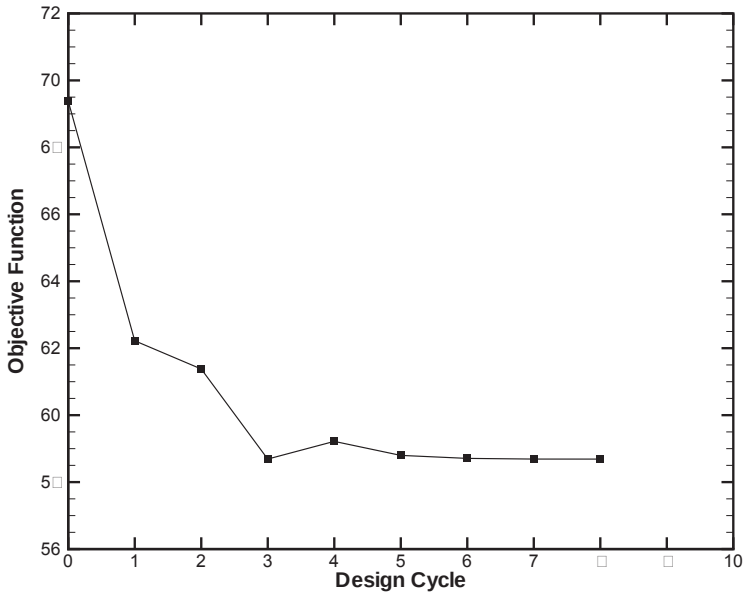


Figure 8. Convergence of objective function for wind turbine case.

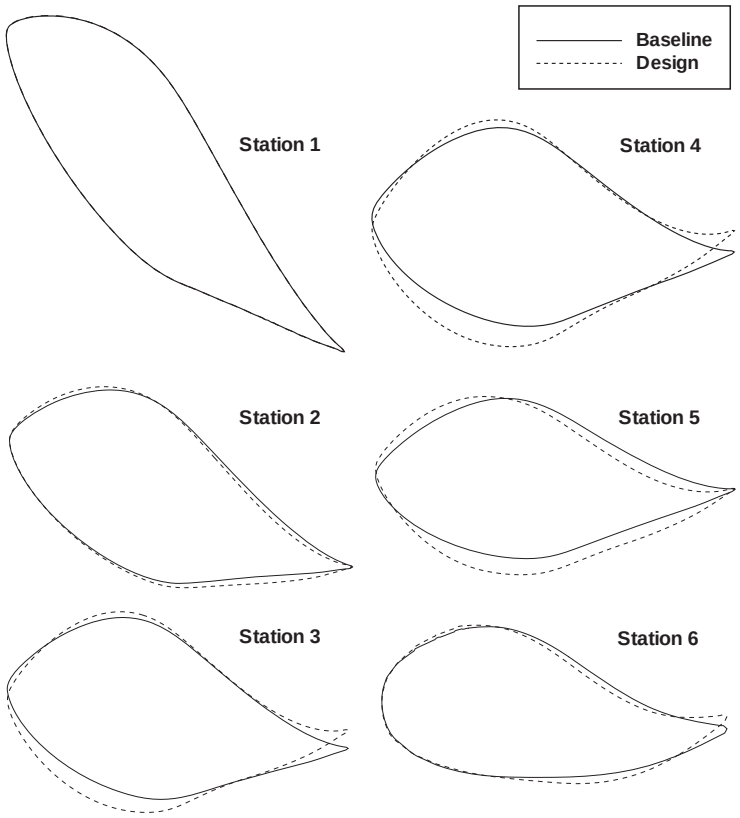


Figure 9. Baseline and final blade section geometries for the wind turbine configuration. Vertical scale has been exaggerated for clarity.

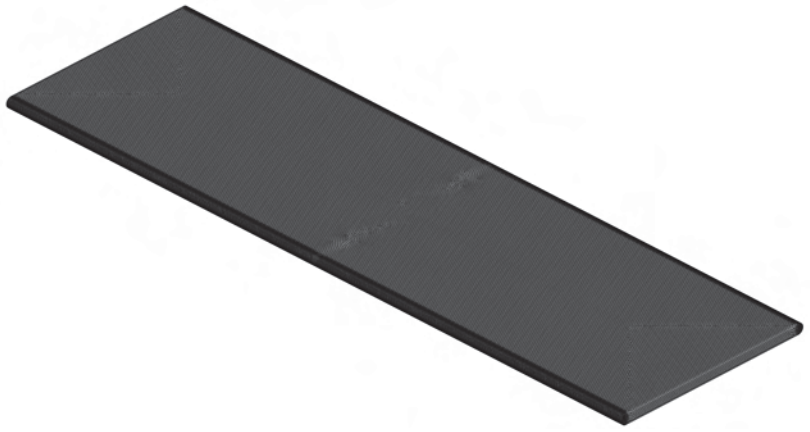
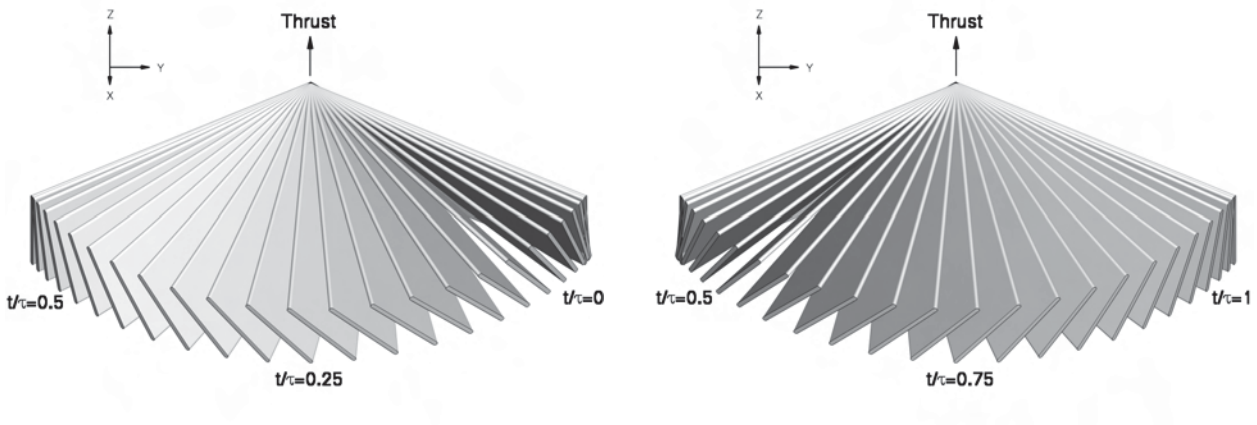


Figure 10. Surface mesh for flapping wing case.



(a) First half of period.

(b) Second half of period.

Figure 11. Snapshots of baseline flapping wing motion.

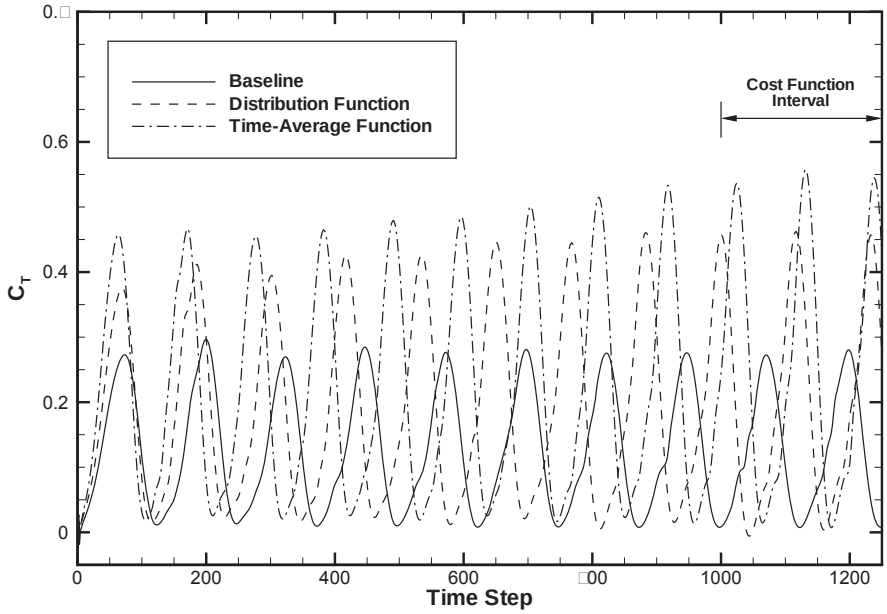


Figure 12. Baseline and final thrust profiles for flapping wing case.

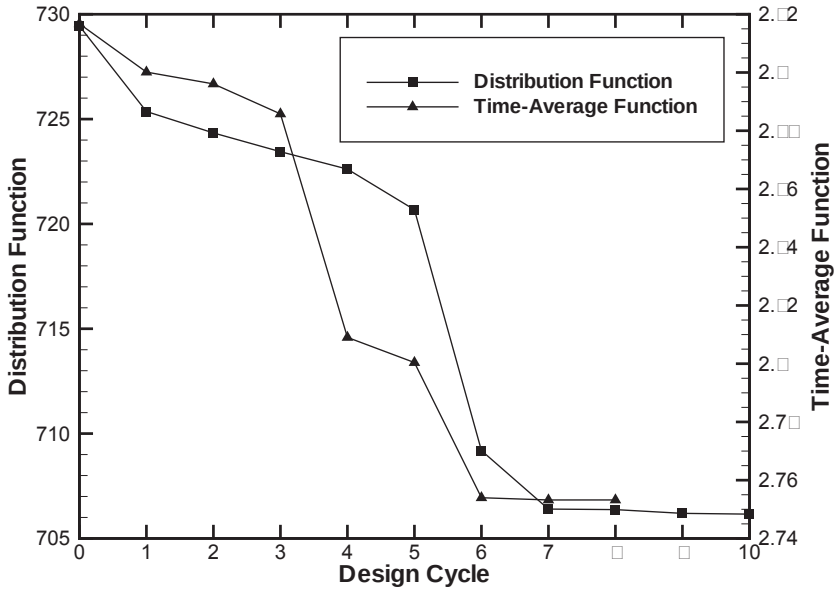


Figure 13. Convergence of objective functions for flapping wing case.

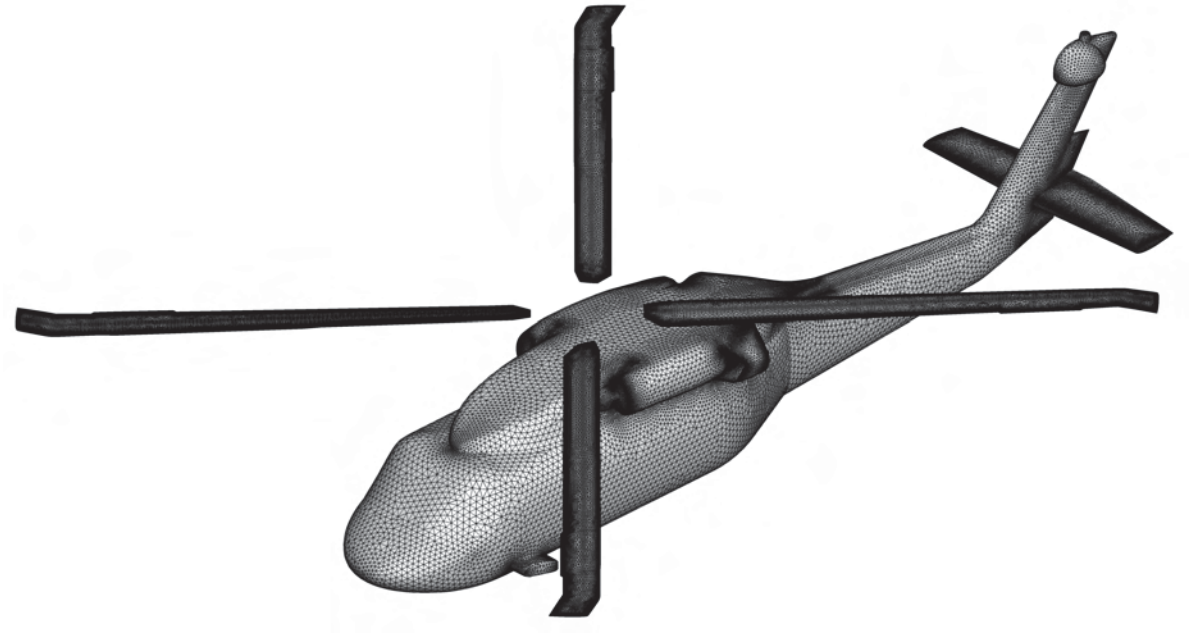


Figure 14. Surface mesh for UH-60 configuration.

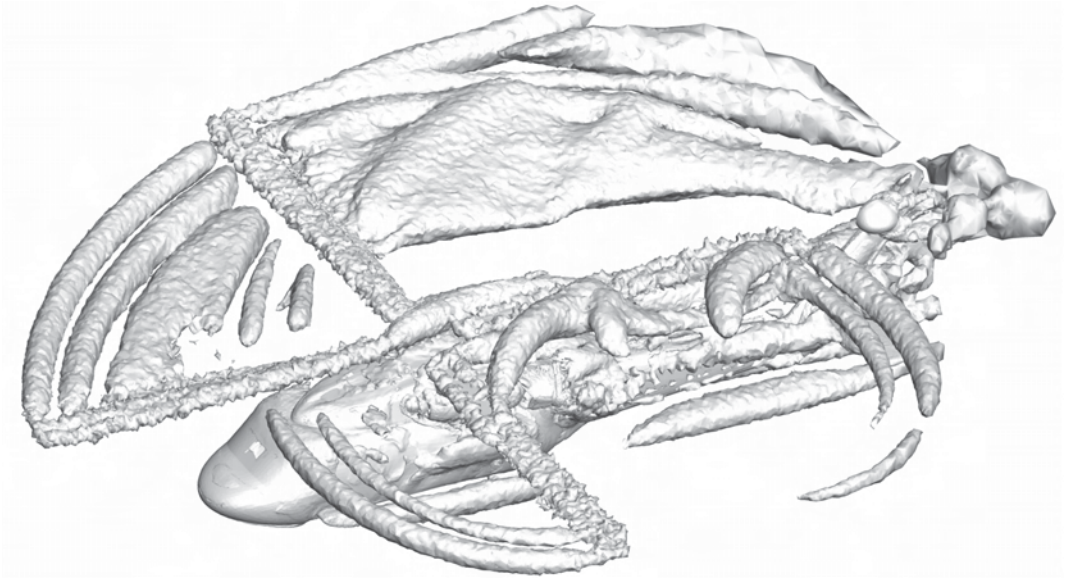


Figure 15. Isosurface of the Q criterion for the baseline UH-60 configuration.

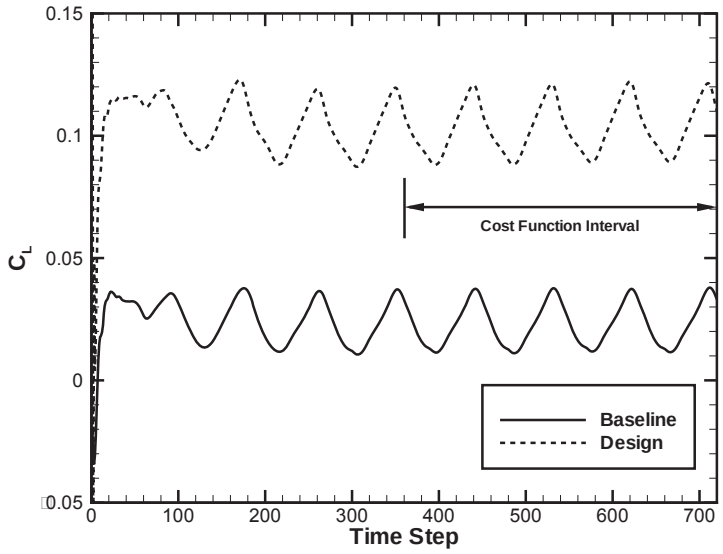


Figure 16. Baseline and final lift coefficient profiles for the UH-60 configuration.

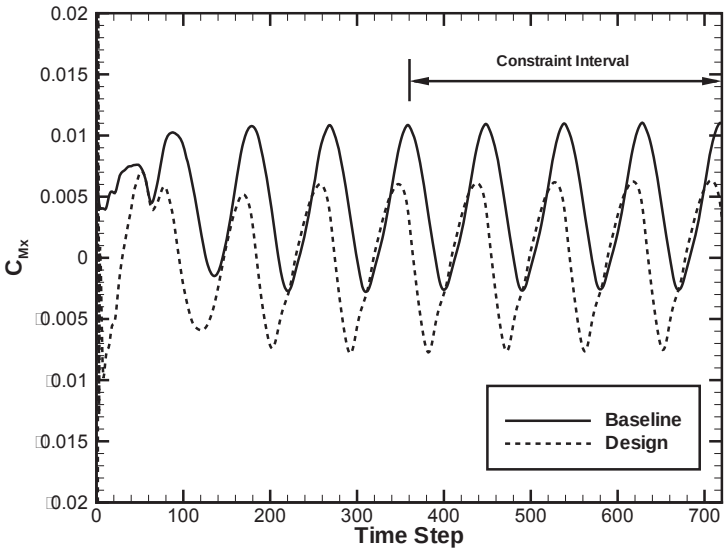


Figure 17. Baseline and final C_{M_x} profiles for the UH-60 configuration.

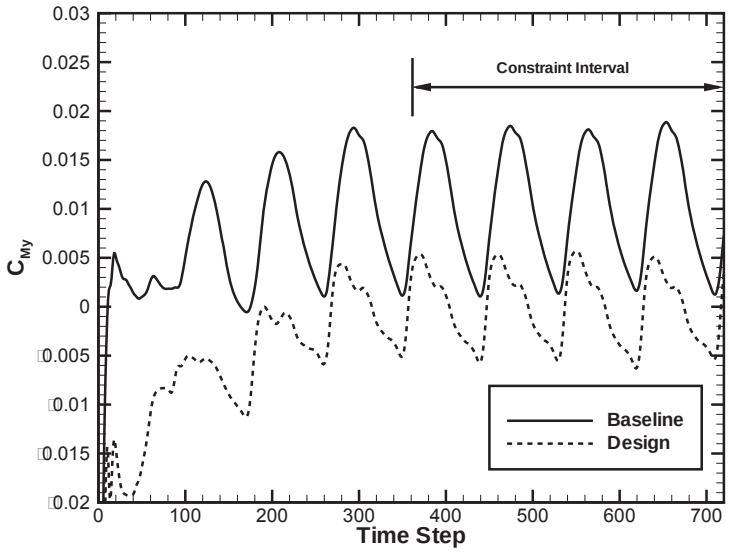


Figure 18. Baseline and final C_{M_y} profiles for the UH-60 configuration.

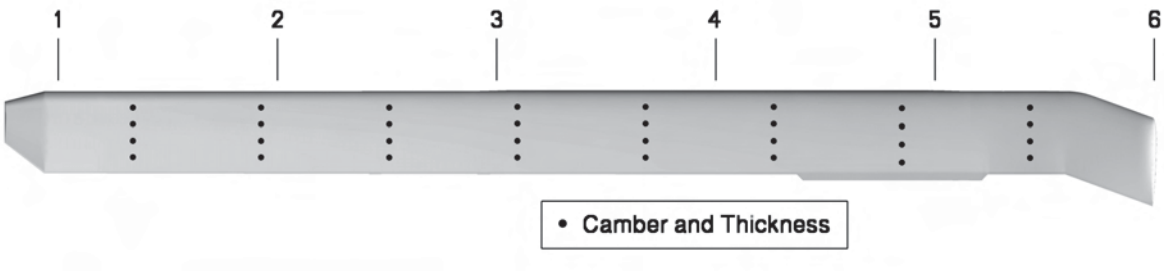


Figure 19. Blade planform geometry, shape variable locations, and spanwise stations for UH-60 configuration.

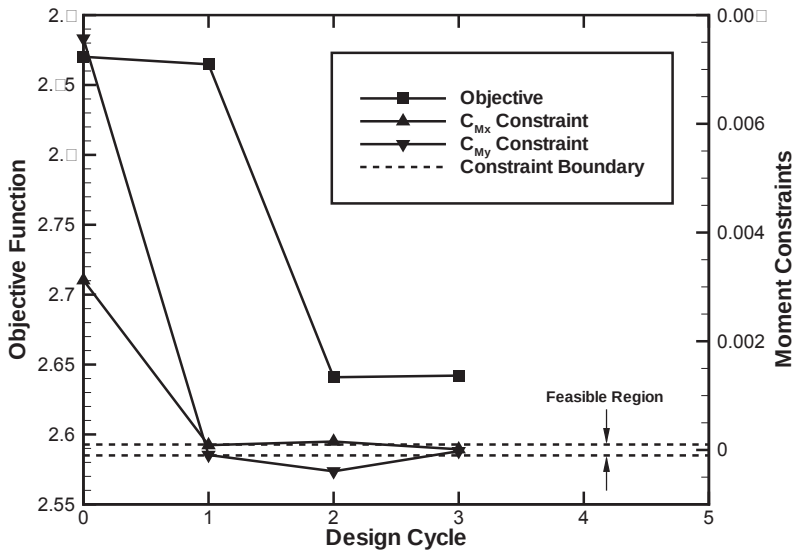


Figure 20. Convergence of the objective function and constraints for the UH-60 configuration.

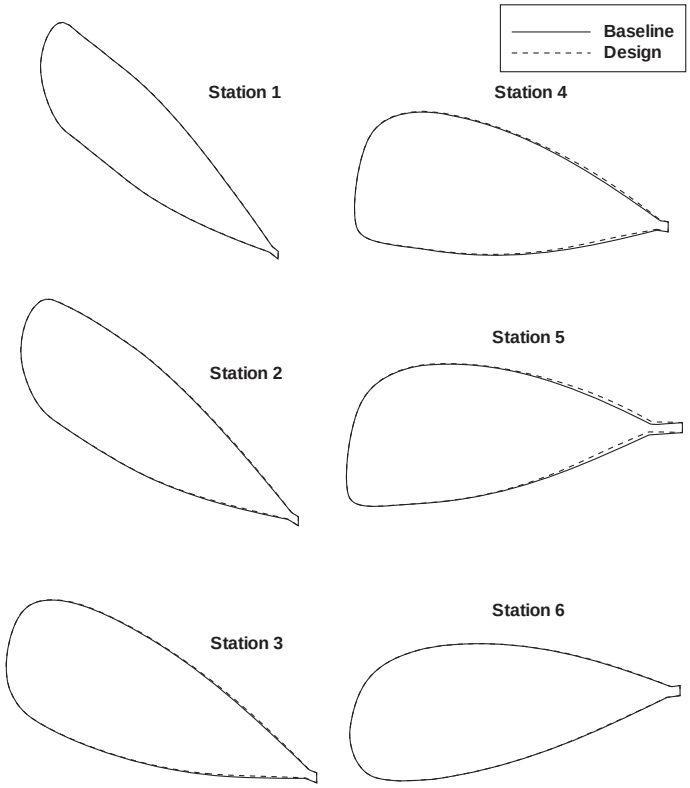


Figure 21. Baseline and final blade section geometries for the UH-60 configuration. Vertical scale has been exaggerated for clarity.

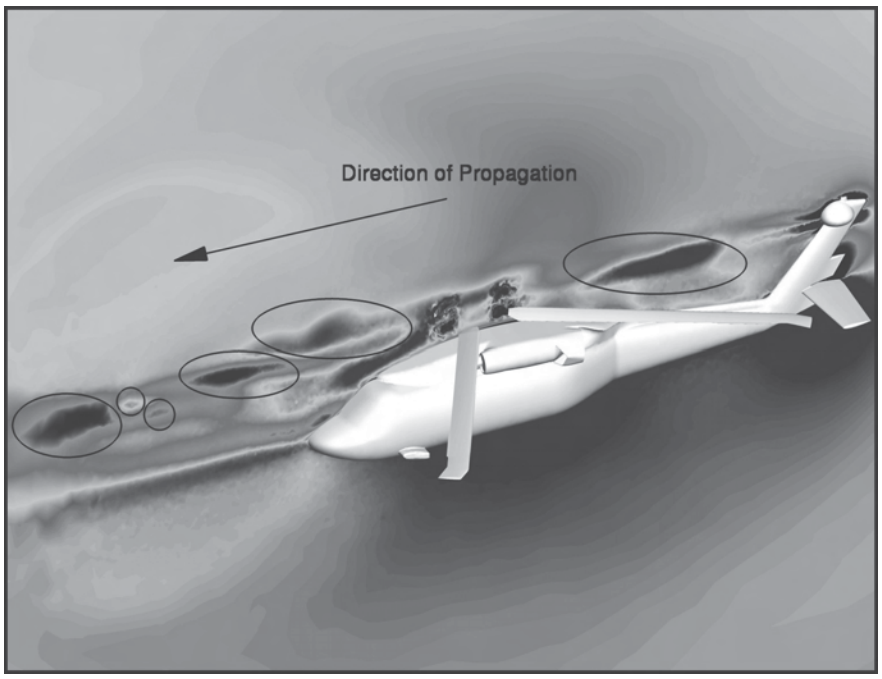


Figure 22. Snapshot of the adjoint solution for the energy equation using an objective function based on a time-averaged lift coefficient. Highlighted features originate on blade surfaces and propagate upstream.

6th AIAA Theoretical Fluid Mechanics Conference
27 - 30 June 2011, Honolulu, Hawaii

AIAA 2011-3111

POD-based Reduced-order Model for Arbitrary Mach Number Flows

K. Pathak* and N. K. Yamaleev†

We develop a new reduced-order model (ROM) based on proper orthogonal decomposition (POD), which can be used for quantitative prediction of not only smooth flows, but also flows with strong discontinuities. In contrast to conventional POD ROMs based on some linearized form of the flow equations, the new model is derived using a Galerkin projection of the original nonlinear discretized 2-D Euler equations onto the POD basis. This approach can be interpreted as a variant of a spectral method with a truncated set of basis functions. A system of nonlinear ODEs obtained this way resembles the major nonlinear and conservation properties of the original discretized Euler equations. The new reduced-order model also preserves the stability properties of the original discrete full-order equations, so that no additional stabilization is required unlike the conventional POD-based models that are inherently unstable. The performance of the new POD ROM is evaluated for 2-D compressible unsteady inviscid flows over a wide range of Mach numbers including trans- and supersonic flows with strong shock waves.

I. Full-order Model

In the present analysis, the dynamics of inviscid compressible flows over a wide range of Mach numbers is described by the time-dependent, two-dimensional Euler equations written in an integral conservation law form as follows:

$$\frac{\partial(\mathcal{V}\mathbf{Q})}{\partial t} + \oint_{\Gamma} \mathbf{F}\cdot\mathbf{n}d\Gamma = \mathbf{0}, \tag{1}$$

where $\mathcal{V}$ is a control volume, $\mathbf{n}$ is the outward unit face normal vector of the control volume with boundary Γ, $\mathbf{Q}$ is the vector of conservative variables averaged over the control volume. The inviscid flux vector $\mathbf{F}$ in Eq. (1) is given by

$$\mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho v u \\ (E+p)u \end{bmatrix} \mathbf{i} + \begin{bmatrix} \rho v \\ \rho v^2 + p \\ \rho u v \\ (E+p)v \end{bmatrix} \mathbf{j} \tag{2}$$

The time derivative and contour integral in Eq. (1) are discretized by a 2nd-order backward difference (BDF2) formula and 2nd-order node-centered finite volume scheme,[1] respectively. The control volume around each grid node is constructed by connecting the centroids of the primal-mesh cells with midpoints of the surrounding edges. The discretized Euler equations including the boundary conditions can be written as follows:

$$\begin{aligned} &V\tfrac{3\mathbf{Q}^n - 4\mathbf{Q}^{n-1} + \mathbf{Q}^{n-2}}{2\Delta t} + \mathbf{R}(\mathbf{Q}^n) = \mathbf{0}, \quad \text{for } 2 \le n \le N_t \\ &V\tfrac{\mathbf{Q}^1 - \mathbf{Q}^0}{\Delta t} + \mathbf{R}(\mathbf{Q}^1) = \mathbf{0}, \qquad\qquad \text{for } n = 1, \end{aligned} \tag{3}$$

where N_t is the total number of time steps, V is a diagonal matrix composed of individual control volumes, and $\mathbf{R}^n$ is a spatial undivided residual approximating the contour integral in Eq. (1).

*Postdoctoral Research Associate, North Carolina A&T State University.
†Corresponding author, Associate Professor, North Carolina A&T State University, E-mail: nkyamale@ncat.edu. Senior member AIAA.

Copyright © 2011 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

The discrete flux $\hat{\mathbf{F}}$ used for approximating the contour integral in Eq. (3) is computed using Roe's approximate Riemann solver[2]

$$\hat{\mathbf{F}} = \frac{1}{2}\left[\hat{\mathbf{F}}_L + \hat{\mathbf{F}}_R - |\mathbf{A}|\left(\mathbf{Q}_L - \mathbf{Q}_R\right)\right], \tag{4}$$

where $\hat{\mathbf{F}}_L$ and $\hat{\mathbf{F}}_R$ are the "left" and "right" normal fluxes at the edge midpoint, $\mathbf{Q}_L$ and $\mathbf{Q}_R$ are the "left" and "right" reconstructed values of the solution vector at the edge midpoint, obtained from some polynomial approximation defined on each control volume, $|\mathbf{A}|$ is the Roe averaged matrix.

Though only the second-order backward-difference (BDF2) formula is used in the present analysis, other high-order BDF and Runge-Kutta schemes can readily be incorporated in the current formulation with minor modifications.[3,4] At each time level, the system of discretized flow equations (3) is solved using the Newton method. To compute the Jacobian matrix required for the Newton solver, the complex-variable approach is employed,[6] which provides discretely exact values of the Jacobian for sufficiently small values of the imaginary step size.

II. POD-based Reduced-Order Model

II.A. Proper Orthogonal Decomposition

We use a method of snapshots developed by Sirovich[5] for constructing the discrete proper orthogonal decomposition basis, also known as the Karhunen-Loéve basis. This approach has been successfully used for incompressible[7] and weakly compressible, subsonic[8] flows, for which the POD reduced-order model is constructed either only for the velocity field or the velocity and pressure fields. For highly compressible flows, this simplified formulation is inadequate, and a proper reduced-order model should include equations for all conservative variables. Furthermore, ROM should preserve the major nonlinear and conservation properties of the original discrete governing equations, which is critical for accurate simulation of flows with shock waves and contact discontinuities. To achieve this goal, we construct its own set of POD basis functions for each conservative variable. This approach is presented next.

The key idea of the proper orthogonal decomposition can be formulated as follows. For a given collection of M snapshots $\{\mathbf{Q}^{n_1}, \cdots, \mathbf{Q}^{n_M}\}$ (elements in a vector space), find a subspace of fixed, much smaller dimension, which is optimal in the sense that the error in the projection onto the subspace is minimized in the L_2 sense. As has been shown in Ref.[5], this constrained optimization problem reduces to a discrete eigenvalue problem. Applying this approach to a kth component of the vector conservative variables $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4]^T$, where the dimensionality of each vector $\mathbf{q}_k$, $1 \le k \le 4$ is equal to the total number of grid points N_g, leads to the following eigenvalue problem:

$$C^k U^k = U^k \Lambda^k, \text{ for } 1 \le k \le 4, \tag{5}$$

where U^k is a matrix of eigenvectors of C^k, and Λ^k is the corresponding diagonal matrix of eigenvalues. The $M \times M$ correlation matrix C^k for the kth component of the vector of conservative variables is given by

$$c_{ij}^k = \frac{1}{M}\langle \mathbf{q}_k^{n_i}, \mathbf{q}_k^{n_j} \rangle, \text{ for } 1 \le k \le 4,\ 1 \le i, j \le M \tag{6}$$

where $\mathbf{q}_k^{n_i}$ is the ith snapshot of the kth conservative variable. The inner product $\langle \cdot, \cdot \rangle$ in Eq. (6) is defined as

$$\langle \mathbf{w}, \mathbf{v} \rangle = \sum_{l=1}^{N_g} V_l w_l v_l, \tag{7}$$

where N_g is the total number of grid points, and V_l is the lth control volume. The POD basis functions for the kth conservative variable are then computed as a linear combination of the snapshot basis functions, whose coefficients are components of the corresponding eigenvector of the correlation matrix C^k

$$\boldsymbol{\psi}_i^k = \sum_{j=1}^{M} u_{ij}^k \mathbf{q}_k^{n_j}, \text{ for } 1 \le i \le M,\ 1 \le k \le 4, \tag{8}$$

where $\boldsymbol{\psi}$ is a vector of length N_g, $\mathbf{q}_k^{n_j}$ is the jth snapshot of the kth conservative variable, u_{ij}^k is the jth component of the ith eigenvector associated with the kth conservative variable, i.e., the ij-th element of the matrix U^k. Since the POD basis functions are nothing else as a linear combination of the flow solution

snapshots, they inherit the major properties of the original data. For example the POD basis function satisfy the boundary conditions of the discrete scheme used for computation of the snapshot basis.

Each correlation matrix C^k is symmetric and positive semidefinite, so that its eigenvalues are all real and non-negative. The corresponding POD basis $\left\{\boldsymbol{\psi}_1^k, \cdot, \boldsymbol{\psi}_M^k\right\}$ is orthogonal and normalized, so that

$$\langle \boldsymbol{\psi}_i^k, \boldsymbol{\psi}_j^k \rangle = \delta_{ij}, \tag{9}$$

where δ_{ij} is the Kronecker delta. The ith eigenvalue λ_i^k of the correlation matrix C^k represents averaged "energy" captured by ith POD mode $\boldsymbol{\psi}_i^k$, where the energy is defined in the sense of the inner product given by Eq. (7) (e.g., see Ref. [5]). For many practical applications, the eigenvalues decay very rapidly, so that a very small number of POD modes $m \ll M$ is sufficient to capture most of the "energy" in the snapshot basis. In the present study, a fixed number of POD modes (typically 5) are used to model the flow dynamics. The first m POD modes capture $\sum_{j=1}^m \lambda_j^k / \sum_{j=1}^M \lambda_j^k$ percentage of the total energy associated with the snapshot basis of the kth conservative variable. Unlike the conventional POD basis that is constructed for deviation of full-order discrete solution $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4]^T$ from the mean of the ensemble $\bar{\mathbf{Q}} = \sum_{j=1}^M \mathbf{Q}^{n_j}/M$, the proper orthogonal decomposition outlined above is based on the snapshot basis itself. The present POD ROM does not require linearization of the Euler equations about the mean flow $\bar{\mathbf{Q}}$, which makes this approach applicable for modeling flows over a wide range of Mach numbers. Also, note that the above POD methodology can be directly used for both structured and unstructured grid formulations.

To preserve the major properties of the original nonlinear full-order model, each conservative variable is expanded separately in its own set of POD modes:

$$\mathbf{q}_k^n \approx \hat{\mathbf{q}}_k^n = \sum_{j=1}^m a_{kj}^n \boldsymbol{\psi}_j^k, \text{ for } 1 \le k \le 4, \tag{10}$$

where $\mathbf{q}_k^n$ and $\hat{\mathbf{q}}_k^n$ are the full- and reduced-order solutions of the kth conservative variable at the nth time level, and a_{kj}^n are the corresponding modal coefficients that depend only on time. The above approach can be interpreted as a spectral method with the truncated set of basis functions $\left\{\boldsymbol{\psi}_j^k\right\}_{j=1}^m$. The full set of POD modes $\left\{\boldsymbol{\psi}_j^k\right\}_{j=1}^M$ for $1 \le k \le 4$ is complete in the sense that any realization contained in the original set of snapshots can be recovered exactly. Note, however, that the truncated set of POD modes $\left\{\boldsymbol{\psi}_j^k\right\}_{j=1}^m$, which is used in Eq. (10), is incomplete and therefore introduces an error in the reduced-order solution. The optimality of the POD basis in the "energy" sense suggests that the truncated set of POD modes is sufficient to accurately describe the full-order solution over that interval of time from which the POD snapshots have been obtained.

II.B. Galerkin Projection

We derive a reduce-order model by using a Galerkin projection of the discretized Euler equations (3) onto the POD basis constructed in the foregoing section. Note that the conventional POD ROMs use some linearized form of the governing equations for derivation of ODEs for the modal coefficients. As a result, this approach is not applicable to discontinuous flows and would lead to wrong prediction of the shock position and its strength. To overcome this problem, we project the original discrete equations (3), which are obtained using the fully conservative finite volume scheme, onto the POD basis. Substituting the expansions (10) into the discretized Euler equations (3), taking inner products with the corresponding POD modes, and using the orthogonality of the POD basis functions lead to the following system of nonlinear ODEs:

$$\frac{3\mathbf{a}^n - 4\mathbf{a}^{n-1} + \mathbf{a}^{n-2}}{2\Delta t} + \hat{\mathbf{R}}(\hat{\mathbf{Q}}^n) = \mathbf{0}, \tag{11}$$

where $\mathbf{a}^n = [\mathbf{a}_1^n, \mathbf{a}_2^n, \mathbf{a}_3^n, \mathbf{a}_4^n]^T$. Components of the reduced-order residual vector $\hat{\mathbf{R}} = [\hat{\mathbf{R}}_1, \hat{\mathbf{R}}_2, \hat{\mathbf{R}}_3, \hat{\mathbf{R}}_4]^T$ are given by

$$\hat{r}_{kj} = \langle \mathbf{R}_k, \boldsymbol{\psi}_j^k \rangle \text{ for } 1 \le k \le 4,\ 1 \le j \le m, \tag{12}$$

where $\hat{\mathbf{R}}_k = [\hat{r}_{k1}, \cdots, \hat{r}_{km}]^T$ and $\mathbf{R}_k$ is the full-order model residual associated with the kth conservative variable. Initial conditions for the nonlinear ODEs (11) are obtained by projecting the initial condition of the full-order model $\mathbf{Q}^0 = [\mathbf{q}_1^0, \mathbf{q}_2^0, \mathbf{q}_3^0, \mathbf{q}_4^0]^T$ onto the POD basis

$$a_{kj}^0 = \langle \mathbf{q}_k^0, \boldsymbol{\psi}_j^k \rangle, \text{ for } 1 \le k \le 4, \ 1 \le j \le m. \tag{13}$$

The equations (11–13) represent a reduced-order model of the discretized Euler equations (3). The original system (3) consisting of $4N_g$ equations has been reduced to a system of $4m$ coupled nonlinear ODEs, where the number of POD modes m used for each conservative variable is much smaller than the total number of grid points N_g. Typically, the number of POD modes required to capture a large portion of the "energy" in the system is of the order of $O(10)$, while the typical number of grid points used in our 2-D simulations is $O(10^4)$, thus providing three orders of magnitude reduction in the number of degrees of freedom used for modeling the flow dynamics. Note, however, that the actual decrease in the computational cost occurs due to the drastic reduction in the size of the reduced-order model Jacobian matrix as compared with the Jacobian of the original discretized Euler equations, which has to be inverted at each time step of the implicit BDF2 scheme. Along with significant savings in the computational cost, the POD ROM also provides a drastic reduction in the storage cost, which is particularly important for adjoint-based optimization of unsteady flows. Indeed, only m POD modes for each conservative variable should be stored to recover the full system dynamics, whereas the straightforward implementation of a time-dependent adjoint-based optimization method requires the entire flow solution history to be stored for all time levels N_t. Note that for typical unsteady flow simulations, the number of POD modes is much smaller that the total number of time steps used for integration of the full-order model equations, thus drastically reducing the overall storage cost.

III. Stability of POD ROM

It is well know that conventional POD ROMs are usually unstable and require additional stabilization.[7,9] The major sources of this instability include the use of a simplified form of the original governing equations and the lack of dissipation in numerical schemes used for discretizing these equations. Unlike the conventional POD ROMs, the proposed reduced-order model preserves the stability properties of the original full-order model. Let us show that if the full-order model given by Eq. (3) is stable in the sense that all eigenvalues of the linearized discrete operator are located in the left half of the complex plane, then the POD ROM (11) is also stable in the spectral sense. Indeed, assuming that $\mathbf{Q}$ is the exact solution of the semi-discrete flow equations $\mathbf{Q}_t + \mathbf{R}(\mathbf{Q}) = 0$ and $\mathbf{e}$ is a solution error caused by a small perturbation of the initial condition such that $\|\mathbf{e}\| \ll \|\mathbf{Q}\|$, we have

$$\frac{\partial(\mathbf{Q}+\mathbf{e})}{\partial t} + \mathbf{R}(\mathbf{Q}+\mathbf{e}) = 0. \tag{14}$$

Linearizing the above equation with respect to $\mathbf{Q}$ yields

$$\frac{\partial \mathbf{e}}{\partial t} = -\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}\mathbf{e}. \tag{15}$$

For strongly stable numerical schemes, all eigenvalues of the Jacobian matrix $-\partial\mathbf{R}/\partial\mathbf{Q}$ are located in the left half of the complex plane. Therefore, the numerical error does not accumulate during the integration of the full-order model equations. Using a similar approach for a semi-discrete form of the POD ROM equations $\mathbf{a}_t + \hat{\mathbf{R}}(\hat{\mathbf{Q}}(\mathbf{a})) = 0$ leads to:

$$\frac{\partial \hat{\mathbf{e}}}{\partial t} = -\frac{\partial \hat{\mathbf{R}}}{\partial \mathbf{a}}\hat{\mathbf{e}}, \tag{16}$$

where $\mathbf{a} = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4]^T$ and $\hat{\mathbf{R}} = [\hat{\mathbf{R}}_1, \hat{\mathbf{R}}_2, \hat{\mathbf{R}}_3, \hat{\mathbf{R}}_4]^T$ are extended vectors of the modal coefficients and the POD ROM residuals, respectively. Combining the POD basis functions obtained for each conservative variable into a single block-diagonal, $4N_g \times 4m$ matrix, we have

$$\Psi = \begin{bmatrix} \Psi_1 & & & 0 \\ & \Psi_2 & & \\ & & \Psi_3 & \\ 0 & & & \Psi_4 \end{bmatrix}, \text{ with } \Psi_k = \begin{bmatrix} \psi_{11}^k & \cdots & \psi_{1m}^k \\ \vdots & \ddots & \vdots \\ \psi_{N_g 1}^k & \cdots & \psi_{N_g m}^k \end{bmatrix}, \ 1 \le k \le 4. \tag{17}$$

As follows from Eq. (12), the POD ROM residuals are obtained as the inner product of the corresponding flow residual and POD basis function, thus leading to the following relation between the POD ROM and full-order model Jacobians:

$$\frac{\partial \hat{\mathbf{R}}}{\partial \mathbf{a}} = \frac{\partial \left(\Psi^T \mathbf{R}\right)}{\partial \mathbf{a}} = \Psi^T \frac{\partial \mathbf{R}}{\partial \mathbf{Q}} \frac{\partial \mathbf{Q}}{\partial \mathbf{a}} = \Psi^T \frac{\partial \mathbf{R}}{\partial \mathbf{Q}} \Psi, \tag{18}$$

where Ψ is the POD basis matrix defined by Eq. (17). Substituting Eq. (18) into Eq. (16) and multiplying this equation by the matrix Ψ yield

$$\Psi \frac{\partial \hat{\mathbf{e}}}{\partial t} = -\Psi \Psi^T \frac{\partial \mathbf{R}}{\partial \mathbf{Q}} \Psi \hat{\mathbf{e}}, \tag{19}$$

Taking into account that the POD basis functions associated with each conservative variable are orthonormal and independent of time, Eq. (19) is recast as

$$\frac{\partial \left(\Psi \hat{\mathbf{e}}\right)}{\partial t} = -\frac{\partial \mathbf{R}}{\partial \mathbf{Q}} \Psi \hat{\mathbf{e}}, \tag{20}$$

The above equation implies that POD ROM is stable in the spectral sense, because all eigenvalues of $-\partial \mathbf{R}/\partial \mathbf{Q}$ are located in the left-half plane, provided that the full-order model equations are stable. Furthermore, comparing Eqs. (20) and (14), one can conclude that the POD ROM error is related to the full-order model error as follows:

$$\mathbf{e} = \Psi \hat{\mathbf{e}}. \tag{21}$$

Multiplying the above equation by Ψ^T yields

$$\hat{\mathbf{e}} = \Psi^T \mathbf{e}, \tag{22}$$

thus leading to the following upper bound on the POD ROM error:

$$\|\hat{\mathbf{e}}\| \leq \|\Psi^T\| \|\mathbf{e}\|, \tag{23}$$

where $\|\cdot\|$ is an appropriate norm (e.g., $\|\cdot\|_p$). Since Ψ is a block-diagonal matrix given by Eq. (17), its norm is fully defined by the norms of each block, i.e., $\|\Psi_1\|$, $\|\Psi_2\|$, $\|\Psi_3\|$, and $\|\Psi_4\|$. If the discrete flow problem is well-posed, then the norm of each POD basis function is bounded, thus implying the boundness of $\|\Psi_k\|$ for all k and consequently the boundness of Ψ. The estimate (22) shows that the matrix Ψ plays a role of an amplification operator between the reduced- and full-order model errors. It should also be noted that the estimate (22) becomes exact if the discrete governing equations are linear.

IV. Results and Discussion

The POD ROM presented in Section II is tested on a 2-D inviscid bump flow problem in sub-, trans-, and supersonic regimes. For all test problems considered, the freestream Mach number is given by

$$M(t) = M_0 + \Delta M \cos(\omega t), \tag{24}$$

where ω is set to be $17\pi/9$, so that the period of oscillations T is $18/17$. Since the freestream Mach number oscillates in time, the entire flowfield is unsteady. The bump shape is described by a polynomial satisfying the requirement that its leading and trailing edges continuously meet the straight lower wall on either side of the bump. The bump thickness is set equal to 0.09. The results presented herein are obtained using the 2nd-order node-centered, finite-volume scheme[1] outlined in Section I. All numerical experiments are performed on a 73×25 structured quadrilateral grid. At each time step, the discretized Euler equations and the system of nonlinear ODEs resulted from the POD ROM are solved by Newton's method. The full- and reduced-order Jacobians, which are needed for Newton's method, are computed using the complex variable technique developed by Lyness.[6] The Euler and POD ROM residuals at each time step are driven below 10^{-12}.

The performance and accuracy of the developed POD ROM are evaluated at three different mean inflow Mach numbers, $M_0 = 0.3$, 0.75, and 1.5 which correspond to sub-, trans-, and supersonic flows, respectively. Note that the flow parameters for the trans- and supersonic regimes are chosen so that strong shock waves are present in the flow. For each test problem, the full-order model equations are integrated over 15 periods

of the freestream Mach number oscillations. The correlation matrix given by Eq. (6) is constructed using $M = 40$ snapshots uniformly distributed over the 7th period of the Mach number oscillations. To build the reduced-order model, only the first five ($m = 5$) POD modes for each conservative variable are used, which contain more than 99% of the total "energy" in the system. The POD-based reduced-order model constructed this way is then used to simulate the flow near the bump over next 8 periods of the freestream Mach number oscillations. Note that the time steps used for integration of the full- and reduced-order equations are equal to each other and set to be 1/40th of the period of the inflow Mach number oscillations.

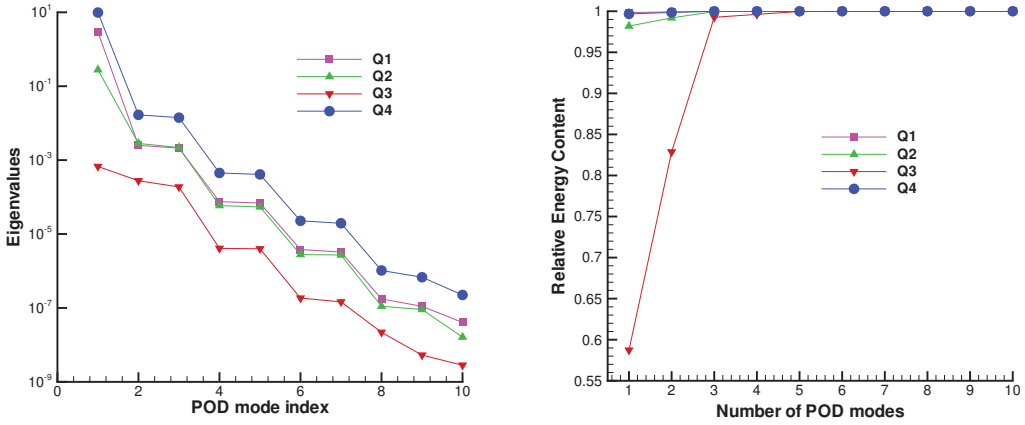


Figure 1. Spectrum of the correlation matrix (left) and relative energy content (right) for each conservative variable for the unsteady subsonic bump flow problem.

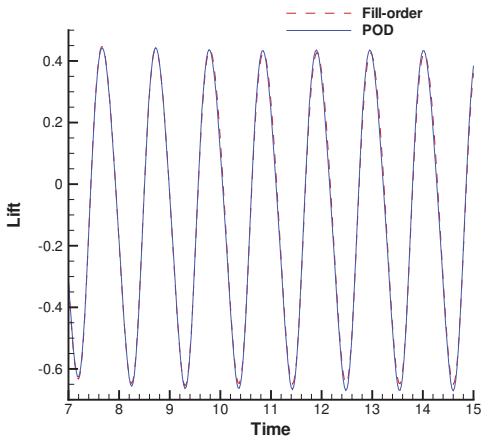


Figure 2. Time histories of the lift coefficient obtained with the full- and reduced-order models for the unsteady subsonic bump flow.

IV.A. Subsonic flow

First, we assess the performance of the developed POD ROM for the subsonic bump flow. The mean Mach number M_0 and the amplitude of oscillations ΔM are set equal to 0.3 and 0.1, respectively. As a result, the flow remains subsonic during the entire time interval considered. To evaluate the efficiency of the POD ROM, the spectra of the correlation matrices for all conservative variables are presented in Fig. 1. For each conservative variable, the eigenvalues rapidly decrease for higher POD modes, thus indicating that only a few first POD basis functions are sufficient to capture nearly the entire energy in the system. Note that

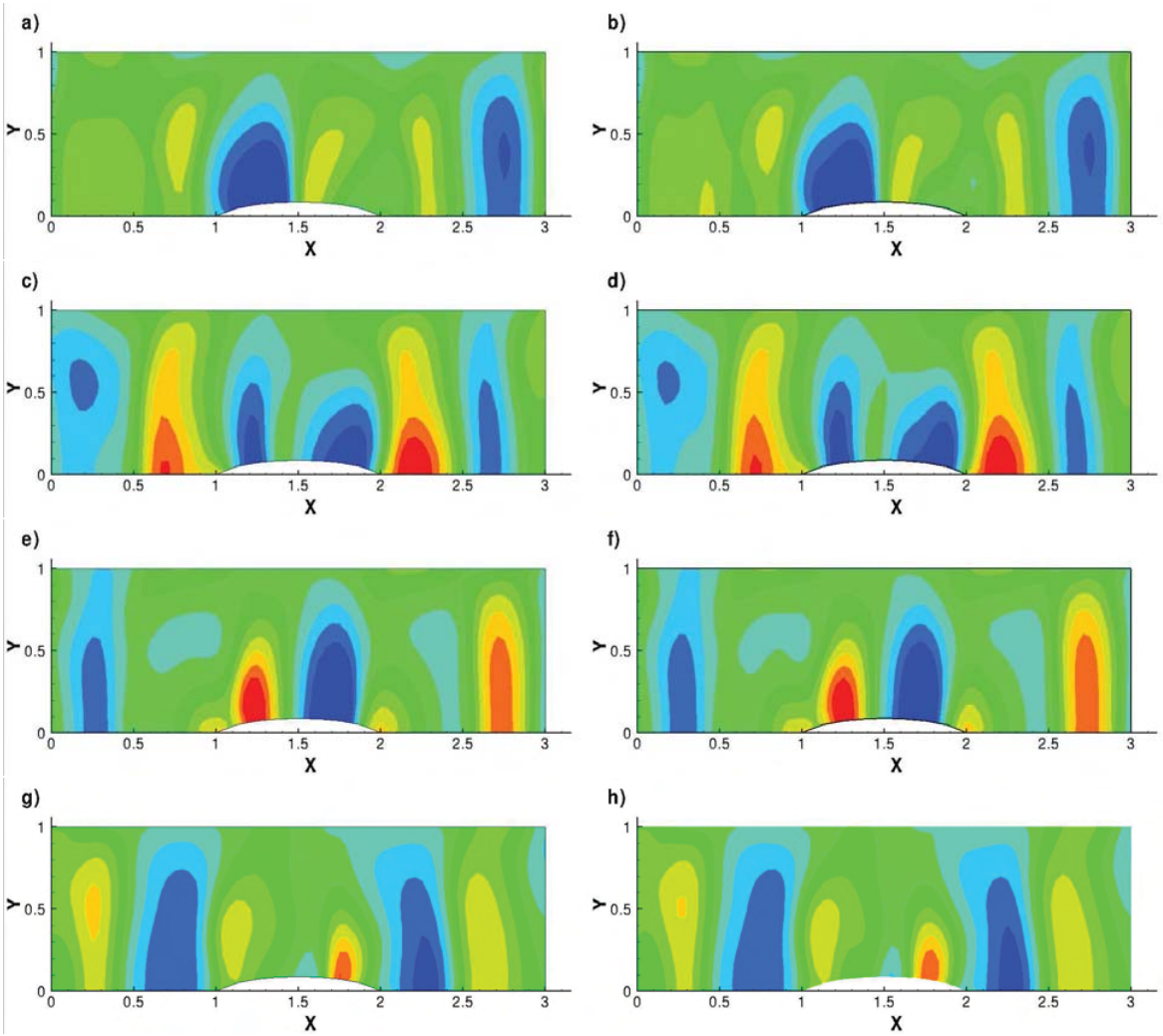


Figure 3. Pressure fields predicted by the full- (left) and reduced-order (right) models at (a-b) $T/8$, (c-d) $3T/8$, (e-f) $5T/8$, and (g-h) $7T/8$ for the unsteady subsonic bump flow.

pairs of eigenvalues corresponding to 2nd and 3rd, 4th and 5th, etc. POD modes are approximately equal to each other, which implies that they make practically the same contribution into the total energy of the system. The relative energy content of each conservative variable, which is given by

$$REC_k = \frac{\sum_{j=1}^{m} \lambda_j^k}{\sum_{j=1}^{M} \lambda_j^k}, \quad k = 1, 2, 3, 4, \tag{25}$$

is also presented in Fig. 1. As follows from this figure, the first 5 POD modes represent nearly 99% of the total energy. Time-histories of the lift coefficient computed using the full- and reduced order models are compared in Fig. 2. The average error in the lift coefficient predicted by the POD ROM is less than 1%, which is consistent with the percentage of "energy" that is not captured by the first 5 POD modes. Figure 3 shows instantaneous pressure fields computed with the full- and reduced-order models at four instants in time $T/8$, $3T/8$, $5T/8$, and $7T/8$ during the 15th period of the freestream Mach number oscillations. The pressure filed is essentially unsteady, which is characterized by the presence of simple waves generated by the inflow Mach number oscillations. As follows from this comparison, the POD ROM can accurately predict not only integral but also local quantities.

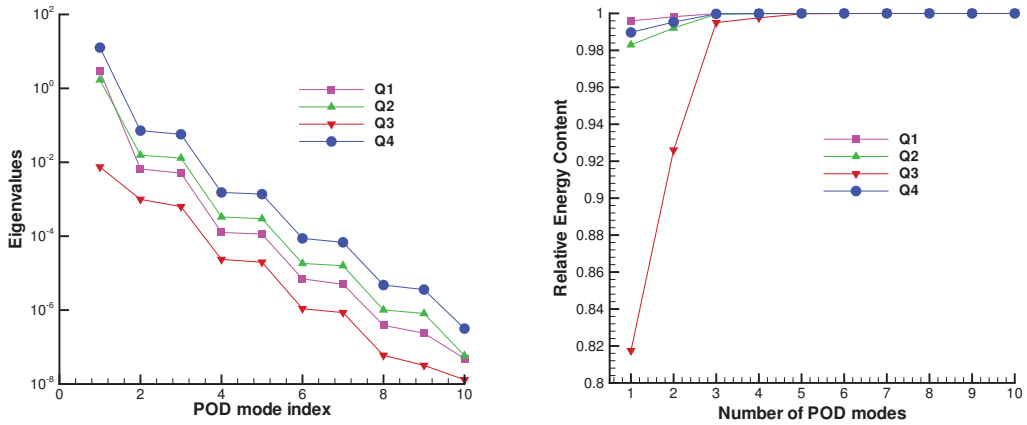


Figure 4. Spectrum of the correlation matrix (left) and relative energy content (right) for each conservative variable for the unsteady transonic bump flow problem.

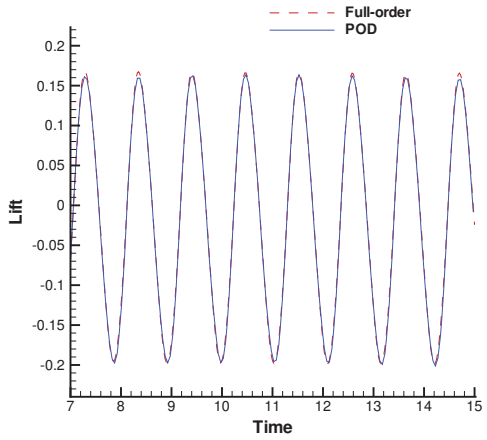


Figure 5. Time histories of the lift coefficient obtained with the full- and reduced-order models for the unsteady transonic bump flow.

IV.B. Transonic flow

For the second test problem, the mean inflow Mach number M_0 and the amplitude of oscillations ΔM are set to be 0.75 and 0.2, respectively. The inflow Mach number is sufficiently high, so that a local supersonic pocket is formed on the bump. The supersonic region is terminated by a shock whose strength and position vary in time during oscillations of the inflow Mach number. The spectrum of the correlation matrix and the relative energy content associated with each conservative variable are shown in Fig. 4. Similar to the subsonic case, the eigenvalues quickly decay as a POD mode index increases. Note, however, that this decay is not as fast as the one obtained in the subsonic case. As a result, the relative energy content of each conservative variable in the transonic case has a boundary layer profile that is wider than that obtained for the subsonic flow. The new POD ROM demonstrates high efficiency because only 5 POD modes capture more than 99% of total energy in the system. In spite of the presence of the shock wave in the computational domain, the lift coefficient predicted by the POD ROM is in excellent agreement with that of the full-order model, as seen in Fig. 5. Another distinctive feature of the new POD ROM is that it very accurately predicts the entire unsteady pressure field, as one can see in Fig. 6. As follows from this comparison, the position and strength of the transonic shock wave computed with the POD ROM agree very well with the solution of the 2-D unsteady Euler equations. The main reason for such a behavior is the fact that the POD ROM

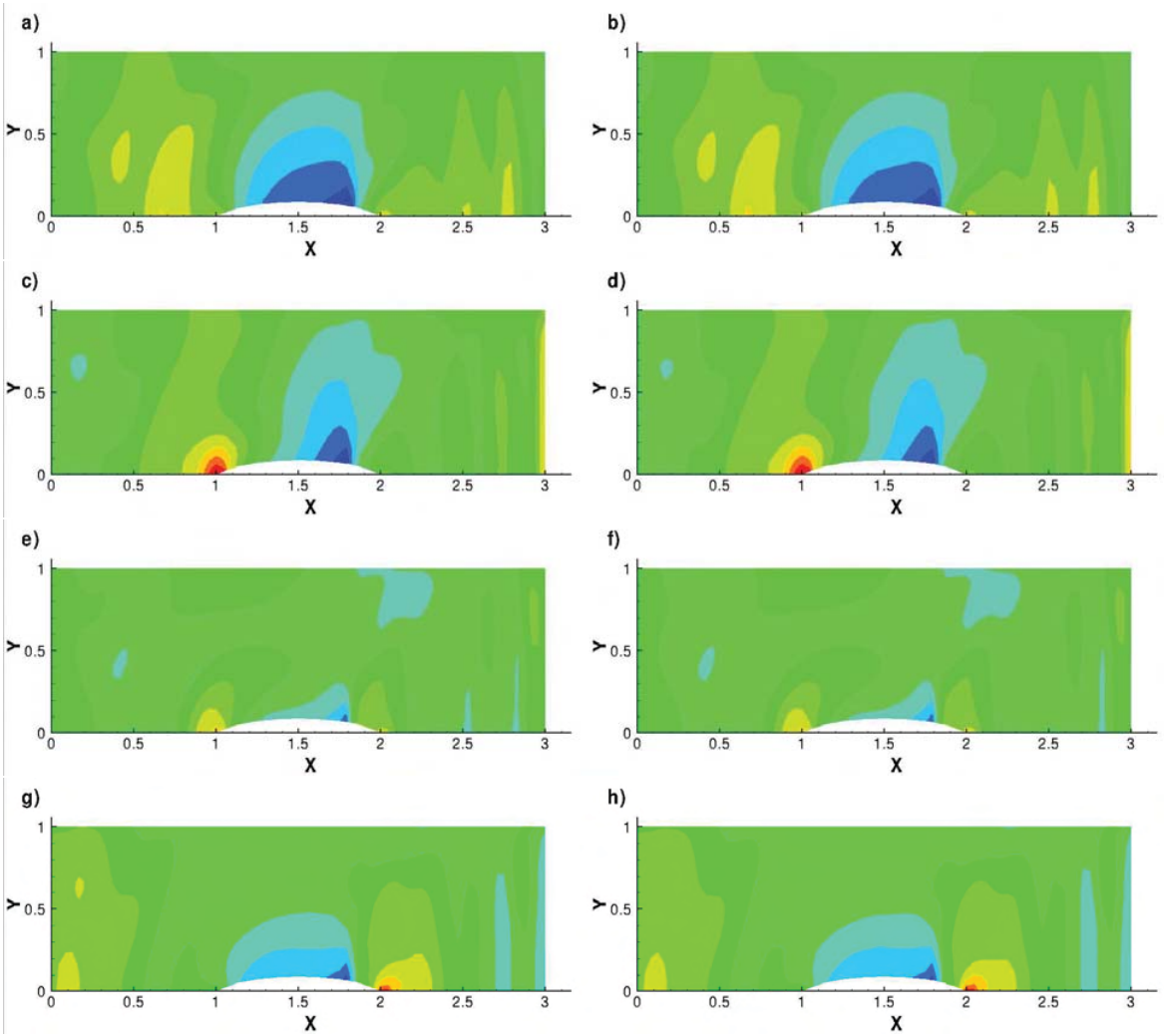


Figure 6. Pressure fields predicted by the full- (left) and reduced-order (right) models at (a-b) $T/8$, (c-d) $3T/8$, (e-f) $5T/8$, and (g-h) $7T/8$ for the unsteady transonic bump flow.

equations are derived using the Galerkin projection of the fully conservative finite volume discretization of the Euler equations, thus preserving the major nonlinear features of the original full-order model.

IV.C. Supersonic flow

The last test problem considered is the unsteady supersonic inviscid flow near the same bump geometry used in the previous test problems. The mean Mach number and the amplitude of oscillations are set equal to 1.5 and 0.3, accordingly. Since the flow is supersonic in the entire domain, two oblique shock waves are formed at both ends of the bump. Note that the leading-edge shock is stronger than the trailing-edge shock which is not well resolved on the 73×25 mesh. Figure 7 shows eigenvalues of the correlation matrices and the relative energy contents associated with all conservative variables. The presence of the strong discontinuities in the flow have no significant effect on the rate of decay of the eigenvalues of the correlation matrix. Similar to the previous cases, the first 5 POD modes contain 99% of the total energy, which gives us an indication that the proposed ROM is capable of efficiently simulate not only sub- and transonic flows, but also supersonic flows with strong shock waves. In contrast to the conventional POD ROMs that are linear in nature and cannot therefore be used for problems with shocks, the present reduced-order model preserves the nonlinear and conservation properties of the original discretized Euler equations. The result is that the developed POD ROM can quantitatively predict both integral and local flow quantities, as one can see in Figs. 8 and 9. As

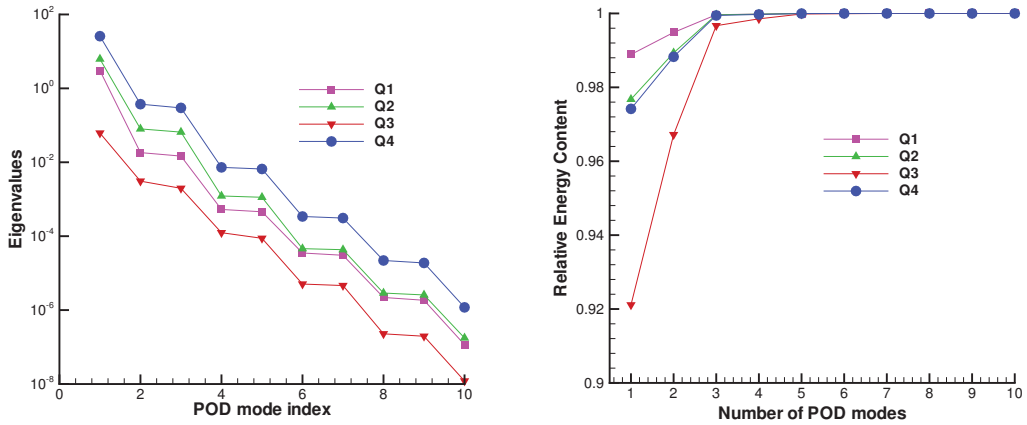


Figure 7. Spectrum of the correlation matrix (left) and relative energy content (right) for each conservative variable for the unsteady supersonic bump flow problem.

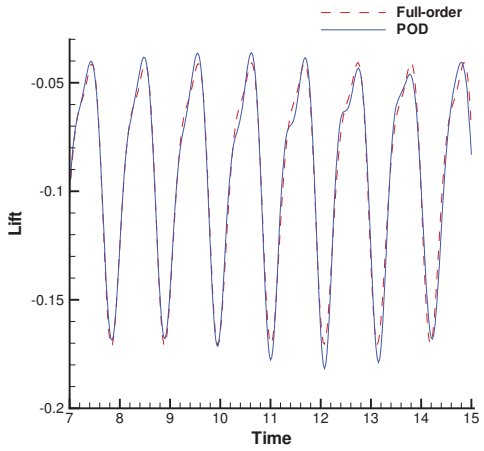


Figure 8. Time histories of the lift coefficient obtained with the full- and reduced-order models for the unsteady supersonic bump flow.

follows from Fig. 8, the agreement between the lift coefficients computed with the full- and reduced-order models is very good. Note that the lift coefficient does not become fully periodic over the time interval considered, which is due to the presence of higher harmonics generated by the shock waves. These results show that the new POD ROM, which is constructed using the snapshots taken only during the 7th period of Mach number oscillations, demonstrates excellent predictive capabilities. Furthermore, the developed POD ROM yields quantitative prediction of the strength and position of the shock waves, as seen in Fig. 9. The results presented in this section suggest that the new POD ROM can be used as an efficient tool for optimization of unsteady compressible flows over a wide range of Mach numbers varying from subsonic to supersonic regimes.

V. Conclusions

A new nonlinear POD-based reduced-order model that is capable of quantitatively predicting continuous and discontinuous flows at arbitrary Mach numbers has been developed and validated. There are two key differences between the new POD ROM and conventional approaches. First of all, the POD basis functions in the new model are constructed for the entire vector of the conservative variables, rather than only for

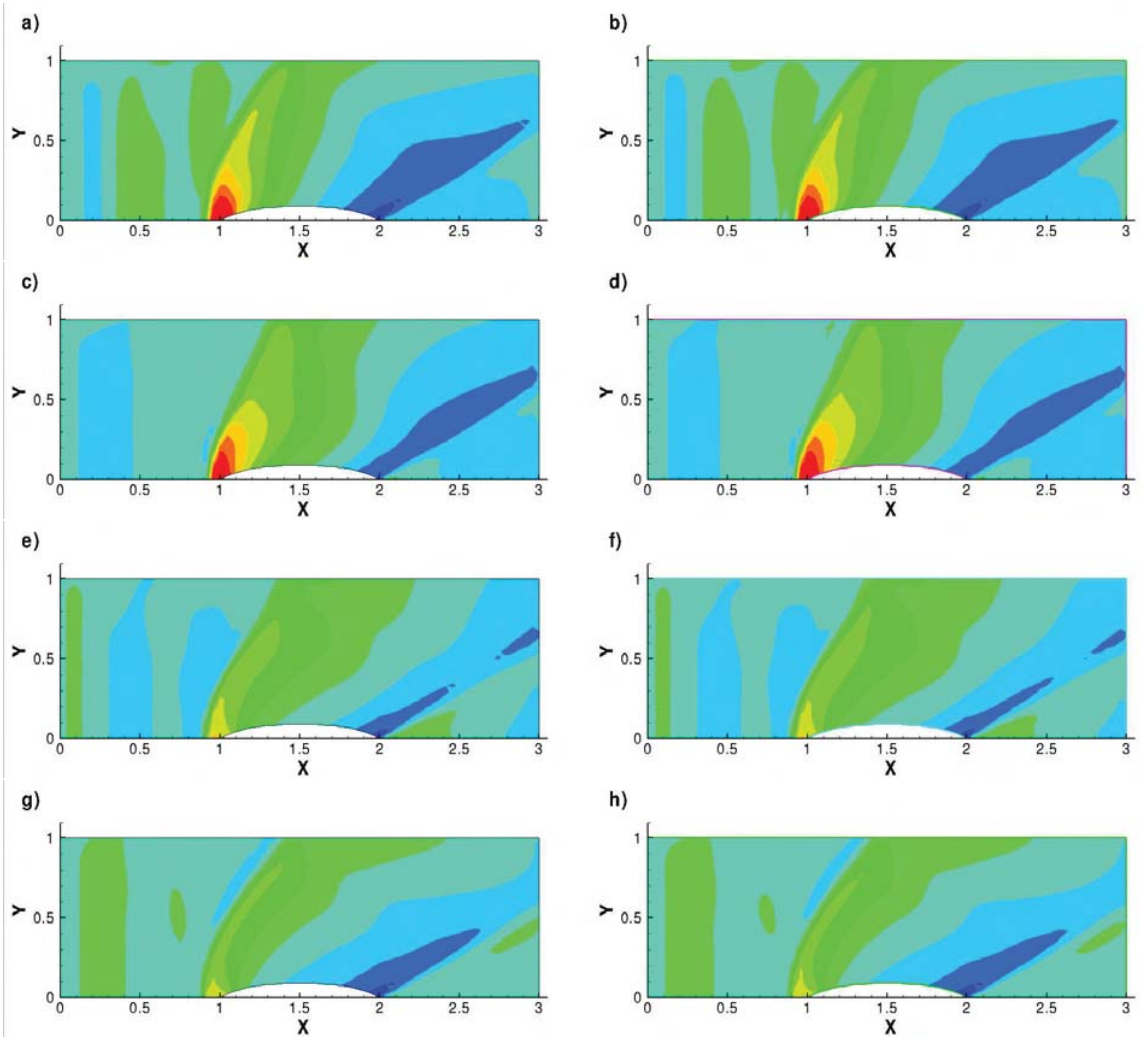


Figure 9. Pressure fields predicted by the full- (left) and reduced-order (right) models at (a-b) $T/8$, (c-d) $3T/8$, (e-f) $5T/8$, and (g-h) $7T/8$ for the unsteady supersonic bump flow.

the velocity field as it is traditionally done in the conventional POD reduced-order models. Secondly, the conventional POD ROMs usually use some linearized form of the original nonlinear discretized or continuous governing equations for derivation of ODEs for the modal coefficients. As a result, these models are not applicable to discontinuous flows and would lead to wrong prediction of the shock position and its strength. In contrast to the approaches available in the literature, the new POD ROM is derived using the Galerkin projection of the original fully conservative discretized Euler equations onto the POD basis, thus leading to a system nonlinear ODEs which closely resembles the nonlinear and conservation properties of the original full-order model. Another attractive feature of the new POD ROM is that it is stable if the numerical scheme associated with the discrete full-order model is stable in the sense that all eigenvalues of the corresponding Jacobian matrix are located in the left half of the complex plane. As a consequence of this, no additional stabilization terms are introduced into the new ROM unlike the conventional POD-based models that require additional dissipation to suppress instabilities caused by the inconsistency in the dissipation operators of the reduced and full-order models. The efficiency and accuracy of the new POD ROM have been evaluated for a 2-D inviscid bump flow problem over a wide range of Mach numbers varying from subsonic to supersonic regimes. Our numerical results have shown that only 5 POD modes are sufficient to represent 99% of the total energy in the system, thus demonstrating that the developed ROM is computationally efficient for the test problems considered. Furthermore, the new model quantitatively predicts not only integral quantities,

but also local flow characteristics. The most distinctive feature of the proposed POD ROM is its ability to accurately simulate flows with strong discontinuities. These encouraging results indicate that the new POD reduced-order model can be effectively used for optimization of unsteady compressible flows at arbitrary Mach numbers.

Acknowledgments

The authors gratefully acknowledge the support provided by NASA through Contract NNL07AA23C.

References

[1]Anderson W. K. and Bonhaus D. L., "An implicit upwind algorithm for computing turbulent flows on unstructured grids," *Computers and Fluids*, Vol. 23 pp. 1-21, 1994.

[2]Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Differential Schemes," *J. of Comput. Phys.,* Vol. 43, No. 2, pp. 357-372, 1981.

[3]Nielsen, E. J., Diskin B., and Yamaleev, N. K. "Discrete Adjoint-Based Design Optimization of Unsteady Turbulent Flows on Dynamic Unstructured Grids," *AIAA J.*, Vol. 48, No. 6, 2010.

[4]Kennedy , C. A., and Carpenter, M. H. " Additive Runge-Kutta schemes for convection-diffusion-reaction equations *Appl. Numer. Math.*, Vol. 44, pp. 139-181, 2003.

[5]Sirovich, L., "Turbulence and the dynamics of coherent structures," *Quart. Appl. Math.*, Vol. XLV(3), pp. 561–590, 1987.

[6]Lyness, J. N. and Moler, C. B., "Numerical Differentiation of Analytic Functions," *SIAM J. Numerical Analysis*, Vol. 4, pp. 202-210, 1967. 1967, pp. 124–134.

[7]Bergmann, M., Bruneau, C.-H., Iollo, A., "Enablers for robust POD models," *J. Comput. Phys.*, Vol. 228, pp. 516–538, 2009.

[8]Bourguet, R., Braza, M., Sevrain, A., Bouhadji, A., "Capturing transition around a wing by reduced-order modeling based on compressible Navier-Stokes equations," *Phys. Fluids*, Vol. 21, 094104, 2009.

[9]Lucia, D.J., and Beran, P.S., "Projection methods for reduced order models of compressible flows," *J. Comput. Phys.*, Vol. 188, pp. 252–280, 2003.

AIAA JOURNAL
Vol. 48, No. 6, June 2010

Discrete Adjoint-Based Design Optimization of Unsteady Turbulent Flows on Dynamic Unstructured Grids

Eric J. Nielsen*
NASA Langley Research Center, Hampton, Virginia 23681
Boris Diskin†
National Institute of Aerospace, Hampton, Virginia 23666
and
Nail K. Yamaleev‡
North Carolina Agricultural and Technical State University, Greensboro, North Carolina 27411

DOI: 10.2514/1.J050035

An adjoint-based methodology for design optimization of unsteady turbulent flows on dynamic unstructured grids is described. The implementation relies on an existing unsteady three-dimensional unstructured grid solver capable of dynamic mesh simulations and discrete adjoint capabilities previously developed for steady flows. The discrete equations for the primal and adjoint systems are presented for the backward-difference family of time-integration schemes on both static and dynamic grids. The consistency of sensitivity derivatives is established via comparisons with complex-variable computations. The current work is believed to be the first verified implementation of an adjoint-based optimization methodology for the true time-dependent formulation of the Navier–Stokes equations in a practical computational code. Large-scale shape optimizations are demonstrated for turbulent flows over a tilt-rotor geometry and a simulated aeroelastic motion of a fighter jet.

Nomenclature

a, b, c, d	=	temporal coefficients
C	=	aerodynamic coefficient
C_T	=	rotor thrust coefficient
$\mathbf{D}$	=	vector of design variables
E	=	total energy per unit volume
$\mathbf{F}$	=	flux vector
$\mathbf{F}_i, \mathbf{F}_v$	=	inviscid and viscous flux vectors
f	=	cost function
f, s	=	general functions
$\mathbf{G}$	=	grid operator
i	=	$\sqrt{-1}$
i, j, k, n	=	indices
in	=	quantity at initial conditions
J	=	number of cost function components
$\boldsymbol{K}$	=	$m_x \times m_x$ linear elasticity coefficient matrix
L	=	Lagrangian function
m_q	=	size of vector $\mathbf{Q}$
m_x	=	size of vector $\mathbf{X}$
N	=	number of time levels
$\hat{\mathbf{n}}$	=	outward-pointing normal vector
p	=	pressure, also cost function exponent
$\mathbf{Q}$	=	$m_q \times 1$ vector of volume-averaged conserved variables
$\mathbf{q}$	=	$m_q \times 1$ vector of conserved variables
$\mathbf{R}$	=	$m_q \times 1$ vector of spatial undivided residuals
$\boldsymbol{R}$	=	$m_x \times m_x$ block-diagonal rotation matrix
R	=	3×3 rotation matrix
$\boldsymbol{R}_{\mathrm{GCL}}$	=	$m_q \times m_q$ diagonal geometric conservation law residual matrix
S	=	control volume surface area
T	=	4×4 transform matrix
t	=	time
u, v, w	=	Cartesian components of velocity
V	=	control volume
$\boldsymbol{V}$	=	$m_q \times m_q$ diagonal matrix of cell volumes
$\mathbf{W}$	=	3×1 face velocity vector
$\mathbf{X}$	=	$m_x \times 1$ vector of grid coordinates
$\mathbf{x}$	=	3×1 position vector
x	=	independent variable
x, y, z	=	Cartesian coordinate directions
ε	=	perturbation
Θ	=	rotor blade collective setting
$\boldsymbol{\Lambda}_f$	=	$m_q \times 1$ flowfield adjoint variable
$\boldsymbol{\Lambda}_g$	=	$m_x \times 1$ grid adjoint variable
ρ	=	density
$\boldsymbol{\tau}$	=	3×1 translation vector
$\boldsymbol{\tau}$	=	$m_x \times 1$ translation vector
Ψ	=	rotor azimuth
ω	=	cost function component weight
∞	=	quantity at freestream conditions
$*$	=	target quantity

Introduction

AS COMPUTATIONAL fluid dynamics (CFD) tools become more efficient, accurate, and robust, their role in the analysis and design of new aerospace configurations continues to increase. Computational methods have already become a major integrated component of industrial practices. The use of CFD has been traditionally confined to the steady regime; however, with recent algorithmic improvements and the persistent growth of computational power, CFD methods have begun to make substantial inroads in simulating unsteady flow phenomena. Target applications for these methods are widely abundant; typical examples might include the prediction of aeroelastic characteristics, maneuvering flight conditions, 6 degree-of-freedom simulations, specified motion problems, or flow control simulations, among many others.

Received 30 June 2009; revision received 24 September 2009; accepted for publication 24 October 2009. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0001-1452/10 and $10.00 in correspondence with the CCC.

*Research Scientist, Computational AeroSciences Branch. Senior Member AIAA.

†Associate Fellow; also Visiting Associate Professor, Department of Mechanical and Aerospace Engineering, University of Virginia. Member AIAA.

‡Associate Professor, Department of Mathematics. Member AIAA.

In recent years, steady-state CFD methods have been targeted for use in automated design optimization frameworks. In gradient-based design approaches, one of the major challenges is to obtain sensitivity information for the flowfield at a reasonable cost. Conventional black-box finite difference methods [1] suffer from well-known step-size limitations and incur a computational expense that grows linearly with the number of design variables. Forward, or direct, differentiation methods [2] and techniques based on the use of complex variables [3] mitigate the step-size limitation but still suffer from excessive cost in the presence of many design variables, as is often the case with aerodynamic design applications.

Adjoint methods provide a powerful alternative for aerodynamic sensitivity analysis. In this approach, the sensitivities of an objective function are determined through the solution of an auxiliary, or adjoint, set of equations. Adjoint methods may be further categorized into either continuous or discrete approaches, depending on the order in which the governing equations are differentiated and discretized. One of the features of the discrete approach is that it allows one to account for mesh variation as well; a second adjoint system can be solved to linearize the relationship between the design variables and the mesh operator as described in [4]. The principal advantage of the adjoint approach is that the computational cost is independent of the number of design variables; a rigorous sensitivity analysis for hundreds of variables can be performed at a cost equivalent to the solution of the governing equations themselves. For examples of the use of such methods, see the references cited in [5].

The role of adjoint-based methodologies in mesh adaptation strategies should also be noted. Whereas many traditional mesh adaptation schemes rely on heuristic connections between solution gradient information and local mesh spacing requirements, the adjoint equations establish a rigorous mathematical connection between solution accuracy and the computational grid. The approach has proven quite powerful and has enjoyed success where traditional feature-based approaches have consistently failed. Fidkowski and Darmofal [6] provide a review of recent applications and an extensive list of references on the subject.

Some recent examples of adjoint-based strategies for unsteady aerospace applications are given in [7–14]. The goal of the current work is to extend the time-dependent adjoint formulation for static grids introduced in [14] and the steady-state discrete adjoint capability developed in [4,15–19] to the three-dimensional time-dependent Euler and Reynolds averaged Navier–Stokes equations. The present approach and implementation are valid for unsteady flows on various grids including static grids, dynamic grids undergoing rigid motion, and general morphing grids governed by a mesh deformation scheme based on a linear elasticity analog. This work is believed to be the first verified implementation of an adjoint-based optimization methodology for the true time-dependent formulation of the Navier–Stokes equations in a practical computational code. In the following sections, the unsteady governing equations are presented as well as various mesh motion strategies. These are followed by the derivation of the discrete adjoint equations for the flowfield and mesh, including details concerning their implementation. Examples demonstrating the discrete consistency of the implementation and applications of the design optimization framework to large-scale problems are also shown.

Flowfield Equations

Using the approach outlined in [20], the unsteady Euler and Navier–Stokes equations may be written in the following form for both moving and stationary control volumes:

$$\frac{\partial}{\partial t}\int_V \mathbf{q}\,\mathrm{d}V + \oint_{\mathrm{d}V} (\mathbf{F}_i - \mathbf{F}_v)\cdot\hat{\mathbf{n}}\,\mathrm{d}S = 0 \tag{1}$$

where V is the control volume bounded by the surface $\mathrm{d}V$. The vector $\mathbf{q}$ represents the conserved variables for mass, momentum, and energy, and the vectors $\mathbf{F}_i$ and $\mathbf{F}_v$ denote the inviscid and viscous fluxes, respectively. Note that, for a moving control volume, the inviscid flux vector must account for the difference in the fluxes due to the movement of control volume faces. Given a flux vector $\mathbf{F}$ on a static grid, the corresponding flux $\mathbf{F}_i$ on a moving grid can be defined as $\mathbf{F}_i = \mathbf{F} - \mathbf{q}(\mathbf{W}\cdot\hat{\mathbf{n}})$, where $\mathbf{W}$ is a local face velocity and $\hat{\mathbf{n}}$ is an outward-pointing unit face normal.

By defining a volume-averaged quantity $\mathbf{Q}$ within each control volume,

$$\mathbf{Q} = \frac{\int_V \mathbf{q}\,\mathrm{d}V}{V} \tag{2}$$

the conservation equations take the form

$$\frac{\partial(\mathbf{Q}V)}{\partial t} + \oint_{\mathrm{d}V} (\mathbf{F}_i - \mathbf{F}_v)\cdot\hat{\mathbf{n}}\,\mathrm{d}S = 0 \tag{3}$$

where the conserved variables and inviscid flux vectors are defined as $\mathbf{Q} = [\rho, \rho u, \rho v, \rho w, E]^T$ and

$$\begin{aligned}\mathbf{F}_i = &\begin{bmatrix} \rho(u - W_x) \\ \rho u(u - W_x) + p \\ \rho v(u - W_x) \\ \rho w(u - W_x) \\ (E + p)(u - W_x) + W_x p \end{bmatrix}\hat{\mathbf{i}} \\ &+ \begin{bmatrix} \rho(v - W_y) \\ \rho u(v - W_y) \\ \rho v(v - W_y) + p \\ \rho w(v - W_y) \\ (E + p)(v - W_y) + W_y p \end{bmatrix}\hat{\mathbf{j}} \\ &+ \begin{bmatrix} \rho(w - W_z) \\ \rho u(w - W_z) \\ \rho v(w - W_z) \\ \rho w(w - W_z) + p \\ (E + p)(w - W_z) + W_z p \end{bmatrix}\hat{\mathbf{k}}\end{aligned} \tag{4}$$

The viscous flux vector $\mathbf{F}_v$ is not explicitly shown here. The equations are closed with the perfect gas equation of state and an appropriate turbulence model for the eddy viscosity. Finally, it is worth noting that, for the special case of a spatially and temporally constant state vector, for example, $\mathbf{Q} = (1, 0, 0, 0, 0)^T$, the conservation equations reduce to the geometric conservation law (GCL) [21]:

$$\frac{\partial V}{\partial t} = \oint_{\mathrm{d}V} \mathbf{W}\cdot\hat{\mathbf{n}}\,\mathrm{d}S \tag{5}$$

In computational practice, the discrete GCL residual is added to the flow equations to preserve a constant solution on dynamic grids [20].

The flow solver used in the current work is described in [15,20,22].[§] The code can be used to perform aerodynamic simulations across the speed range, and an extensive list of options and solution algorithms is available for spatial and temporal discretizations on general static or dynamic mixed-element unstructured meshes that may or may not contain overset grid topologies.

In the current study, the spatial discretization uses a finite volume approach in which the dependent variables are stored at the vertices of tetrahedral meshes. Inviscid fluxes at cell interfaces are computed using the upwind scheme of Roe [23], and viscous fluxes are formed using an approach equivalent to a finite element Galerkin procedure. For dynamic mesh cases, the mesh velocity terms are evaluated using backward differences consistent with the discrete time derivative; this makes the spatial and GCL residuals dependent on grids at previous time levels. The eddy viscosity is modeled using the one-

[§]Data available online at http://fun3d.larc.nasa.gov [retrieved 4 January 2010].

equation approach of Spalart and Allmaras [24]. Massively parallel scalability is achieved through domain decomposition and message passing communication.

An approximate solution of the linear system of equations formed within each time step is obtained through several iterations of a multicolor Gauss–Seidel point-iterative scheme. The turbulence model is integrated all the way to the wall without the use of wall functions and is solved separately from the mean flow equations at each time step with a time-integration and linear system solution scheme identical to that employed for the mean flow equations.

Grid Equations

The general grid equations can be defined in the form $\mathbf{G}^n(\mathbf{X}, \mathbf{D}) = 0$, where $\mathbf{X}$ is the mesh (meshes at several time levels may be involved), $\mathbf{D}$ is the vector of design variables, and n denotes the time level and indicates that the grid operator may vary in time. The specific formulations for different grid motions are introduced next.

Grids Undergoing Rigid Motion

For problems in which rigid mesh motion is required, the motion is generated by a 4×4 transform matrix, T, as outlined in [20]. This transform matrix enables general translations and rotations of the grid according to the relation

$$\mathbf{x} = T\mathbf{x}^0 \tag{6}$$

which moves a point from an initial position $\mathbf{x}^0 = (x^0, y^0, z^0)^T$ to its new position $\mathbf{x} = (x, y, z)^T$:

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & \tau_x \\ R_{21} & R_{22} & R_{23} & \tau_y \\ R_{31} & R_{32} & R_{33} & \tau_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^0 \\ y^0 \\ z^0 \\ 1 \end{bmatrix} \tag{7}$$

In an expanded form, $\mathbf{x} = R\mathbf{x}^0 + \boldsymbol{\tau}$. Here, the 3×3 matrix R defines a general rotation and the vector $\boldsymbol{\tau}$ specifies a translation. The matrix T is generally time dependent. One useful feature of this approach is that multiple transformations telescope via matrix multiplication. This formulation is particularly attractive for composite parent–child body motion, in which the motion of one body is often specified relative to another. The reader is referred to the discussion in [20] for more details. For this formulation, the grid operator at time level n is defined as

$$\mathbf{G}^n(\mathbf{X}^n, \mathbf{X}^0, \mathbf{D}) \equiv \boldsymbol{R}^n\mathbf{X}^0 + \boldsymbol{\tau}^n - \mathbf{X}^n \tag{8}$$

where $\mathbf{X}^0$ and $\mathbf{X}^n$ are the grid vectors at the initial and nth time levels, respectively; $\boldsymbol{R}^n$ is an $m_x \times m_x$ block-diagonal matrix with 3×3 blocks representing rotation and m_x being the size of vector $\mathbf{X}^n$; and $\boldsymbol{\tau}^n$ is an m_x-size translation vector. The matrix $\boldsymbol{R}^n$ and vector $\boldsymbol{\tau}^n$ may explicitly depend on $\mathbf{D}$.

Deforming Grids

The simplest example of a deforming grid simulation is a static grid undergoing deformations as a result of a shape optimization process. In this case, the grid is not time dependent and is modeled as an elastic medium that obeys the elasticity relations of solid mechanics. An auxiliary system of linear partial differential equations (PDEs) is solved to determine the mesh coordinates after each shape update. Discretization of these PDEs yields a system of equations

$$\boldsymbol{K}\,\mathbf{X} = \mathbf{X}_{\text{surf}} \tag{9}$$

where $\boldsymbol{K}$ represents the elasticity coefficient matrix, $\mathbf{X}$ is the vector of grid coordinates being solved for, and $\mathbf{X}_{\text{surf}}$ is the vector of updated surface coordinates, complemented by zeros for all interior coordinates.

The coefficients of the matrix $\boldsymbol{K}$ depend on the coordinates of the grid. In the approach followed here, the elasticity equations are discretized on the grid corresponding to the initial time level. Thus, the grid at the initial level satisfies the nonlinear equations

$$\boldsymbol{K}^0(\mathbf{X}^0, \mathbf{D})\mathbf{X}^0 = \mathbf{X}^0_{\text{surf}} \tag{10}$$

The material properties of the system are chosen based on the local cell geometry and proximity to the surface, and the system is solved using a preconditioned generalized minimal residual algorithm. For further details on the approach, see [17,20,25].

For static grid formulations, the only grid operator used at all times is

$$\mathbf{G}(\mathbf{X}, \mathbf{D}) \equiv \mathbf{X}_{\text{surf}} - \boldsymbol{K}\mathbf{X} \tag{11}$$

where $\mathbf{X}_{\text{surf}}$ may explicitly depend on $\mathbf{D}$. There are situations in which time-dependent deforming grids are required, including aeroelastic deflections of the surface, for which the rigid motion as described in the previous section is not valid. Instead, a morphing mesh formulation is used. In this approach, the linear elasticity equations given by Eq. (9) are solved at each time level with the matrix $\boldsymbol{K} = \boldsymbol{K}^0$ computed at the initial time level and fixed throughout the time evolution; the vector $\mathbf{X}^n_{\text{surf}}$ represents the current body positions. For morphing grids, the operator at time level n is defined as

$$\mathbf{G}^n(\mathbf{X}^n, \mathbf{D}) \equiv \mathbf{X}^n_{\text{surf}} - \boldsymbol{K}^0\mathbf{X}^n \tag{12}$$

When the surface motion is governed by the rigid motion relations given by Eq. (6), $\mathbf{X}^n_{\text{surf}}$ can be further specified as $\mathbf{X}^n_{\text{surf}} = \boldsymbol{R}^n\mathbf{X}^0_{\text{surf}} + \boldsymbol{\tau}^n$.

Cost Functions

The steady-state adjoint implementation described in [4,15–19] permits multiple objective functions and explicit constraints of the following form, each containing a summation of individual components:

$$f_i = \sum_{j=1}^{J_i} \omega_j (C_j - C_j^*)^{p_j} \tag{13}$$

Here, ω_j represents a user-defined weighting factor, C_j is an aerodynamic coefficient such as the total drag or the pressure or viscous contributions to such quantities, the superscript * indicates a user-defined target value of C_j, and p_j is a user-defined exponent chosen so that f_i is a convex functional. The user may specify computational boundaries to which each component function applies. The index i indicates a possibility of introducing several different cost functions or constraints, which may be useful if the user desires separate sensitivities, for example, for lift, drag, pitching moment, etc.

For the unsteady formulation, similar general cost functions f_i^n are defined at each time level n. The integrated cost function f_i is defined as a discrete time integral over a certain time interval $[t_i^1, t_i^2]$:

$$f_i = \sum_{n=N_i^1}^{N_i^2} f_i^n \Delta t \tag{14}$$

where time levels N_i^1 and N_i^2 correspond to t_i^1 and t_i^2, respectively. The user now supplies time intervals over which the cost functions are to be used.

Derivation of the Time-Dependent Adjoint Equations

To derive the time-dependent form of the adjoint equations, the methodology developed in [14] is used. The governing equations given by Eq. (3) are rewritten as

$$\frac{\partial(\mathbf{Q}V)}{\partial t} + \mathbf{R} = 0, \qquad \mathbf{R} \equiv \oint_{\mathrm{d}V} (\mathbf{F}_i - \mathbf{F}_v) \cdot \hat{\mathbf{n}}\, \mathrm{d}S \tag{15}$$

Using a first-order backward difference (BDF1) in time, the equations can be evaluated at time level n as follows:

$$\boldsymbol{V}^n \frac{\mathbf{Q}^n - \mathbf{Q}^{n-1}}{\Delta t} + \mathbf{R}^n + \boldsymbol{R}^n_{\mathrm{GCL}} \mathbf{Q}^{n-1} = 0 \tag{16}$$

Here, $\boldsymbol{V}^n$ and $\boldsymbol{R}^n_{\mathrm{GCL}}$ are $m_q \times m_q$ diagonal matrices, m_q is the length of vector $\mathbf{Q}^n$, the GCL is discretized in a consistent fashion as

$$\frac{1}{\Delta t}(\boldsymbol{V}^n - \boldsymbol{V}^{n-1}) = \boldsymbol{R}^n_{\mathrm{GCL}} \tag{17}$$

and $\mathbf{R}^n$ is the spatial undivided residual. Recall that $\mathbf{R}^n$ and $\boldsymbol{R}^n_{\mathrm{GCL}}$ depend on grids at the current and previous time levels. Note also that although the BDF1 scheme has been shown here for the sake of simplicity, the derivations for higher-order temporal schemes are similar and included in the Appendix.

The discrete adjoint-based optimization methodology is based on the method of Lagrange multipliers, which is used to enforce the governing equations as constraints. For the sake of simplicity in the following derivations, a single cost function is assumed; therefore, the index i is omitted. For the time-dependent equations, the Lagrangian functional is defined as follows:

$$\begin{aligned} L(\mathbf{D}, \mathbf{Q}, \mathbf{X}, \boldsymbol{\Lambda}_f, \boldsymbol{\Lambda}_g) = {} & \sum_{n=1}^{N} f^n \Delta t + \sum_{n=1}^{N} [\boldsymbol{\Lambda}_f^n]^T \Bigg(\boldsymbol{V}^n \frac{\mathbf{Q}^n - \mathbf{Q}^{n-1}}{\Delta t} \\ & + \mathbf{R}^n + \boldsymbol{R}^n_{\mathrm{GCL}} \mathbf{Q}^{n-1} \Bigg) \Delta t + \sum_{n=1}^{N} [\boldsymbol{\Lambda}_g^n]^T \mathbf{G}^n \Delta t \\ & + (f^0 + [\boldsymbol{\Lambda}_f^0]^T \mathbf{R}^{\mathrm{in}}) \Delta t + [\boldsymbol{\Lambda}_g^0]^T \mathbf{G}^0 \Delta t \end{aligned} \tag{18}$$

where $f^n \equiv 0$ for $n < N^1$ and $n > N^2$; $\mathbf{G}^n = 0$ are the grid equations at time level n; $\boldsymbol{\Lambda}_f^n$ and $\boldsymbol{\Lambda}_g^n$ are vectors of Lagrange multipliers associated with the flow and grid equations at time level n, respectively; $\mathbf{D}$ is a vector of design variables; and $\mathbf{R}^{\mathrm{in}} = 0$ is the initial condition for the flow equations.

The Lagrangian is differentiated with respect to $\mathbf{D}$, assuming that f^n depends on $\mathbf{Q}^n$, $\mathbf{X}^n$, and $\mathbf{D}$; $\mathbf{R}^{\mathrm{in}}$ depends on $\mathbf{Q}^0$, $\mathbf{X}^0$, and $\mathbf{D}$; $\mathbf{R}^n$ depends on $\mathbf{Q}^n$, $\mathbf{X}^n$, $\mathbf{X}^{n-1}$, and $\mathbf{D}$; and $\boldsymbol{R}^n_{\mathrm{GCL}}$ depends on $\mathbf{X}^n$, $\mathbf{X}^{n-1}$, and $\mathbf{D}$. Regrouping terms to isolate the coefficients of $\partial \mathbf{Q}^n / \partial \mathbf{D}$ and equating the coefficients to zero yields the final form of the adjoint equations for the flowfield:

$$\begin{aligned} & \frac{1}{\Delta t}(\boldsymbol{V}^n \boldsymbol{\Lambda}_f^n - \boldsymbol{V}^{n+1} \boldsymbol{\Lambda}_f^{n+1}) + \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}^n}\right]^T \boldsymbol{\Lambda}_f^n + \boldsymbol{R}^{n+1}_{\mathrm{GCL}} \boldsymbol{\Lambda}_f^{n+1} \\ & \quad = -\left[\frac{\partial f^n}{\partial \mathbf{Q}^n}\right]^T, \quad \text{for } 1 \le n \le N \end{aligned} \tag{19}$$

$$\begin{aligned} & -\frac{1}{\Delta t} \boldsymbol{V}^1 \boldsymbol{\Lambda}_f^1 + \left[\frac{\partial \mathbf{R}^{\mathrm{in}}}{\partial \mathbf{Q}^0}\right]^T \boldsymbol{\Lambda}_f^0 + \boldsymbol{R}^1_{\mathrm{GCL}} \boldsymbol{\Lambda}_f^1 = -\left[\frac{\partial f^0}{\partial \mathbf{Q}^0}\right]^T \\ & \qquad \text{for the initial time level} \end{aligned} \tag{20}$$

where $\boldsymbol{\Lambda}^{N+1} = 0$. Collecting the coefficients of $\partial \mathbf{X}^n / \partial \mathbf{D}$ and equating them to zero leads to similar adjoint equations for the grid. Assuming that the grid operator at time level n, $\mathbf{G}^n$, depends on $\mathbf{X}^n$, $\mathbf{X}^0$, and $\mathbf{D}$, the grid adjoint equations are defined as

$$\begin{aligned} & -\left[\frac{\partial \mathbf{G}^n}{\partial \mathbf{X}^n}\right]^T \boldsymbol{\Lambda}_g^n = \left[\frac{\partial f}{\partial \mathbf{X}^n}\right]^T + \left[\frac{\partial \boldsymbol{V}}{\partial \mathbf{X}^n} \frac{\mathbf{Q}^n - \mathbf{Q}^{n-1}}{\Delta t}\right]^T \boldsymbol{\Lambda}_f^n \\ & \quad + \sum_{k=0}^{1} \left[\frac{\partial \mathbf{R}^{n+k}}{\partial \mathbf{X}^n} + \frac{\partial \boldsymbol{R}^{n+k}_{\mathrm{GCL}}}{\partial \mathbf{X}^n} \mathbf{Q}^{n+k-1}\right]^T \boldsymbol{\Lambda}_f^{n+k}, \quad \text{for } 1 \le n \le N \end{aligned} \tag{21}$$

$$\begin{aligned} & -\left[\frac{\partial \mathbf{G}^0}{\partial \mathbf{X}^0}\right]^T \boldsymbol{\Lambda}_g^0 = \sum_{n=1}^{N} \left[\frac{\partial \mathbf{G}^n}{\partial \mathbf{X}^0}\right]^T \boldsymbol{\Lambda}_g^n + \left[\frac{\partial f^0}{\partial \mathbf{X}^0}\right]^T + \left[\frac{\partial \mathbf{R}^{\mathrm{in}}}{\partial \mathbf{X}^0}\right]^T \boldsymbol{\Lambda}_f^0 \\ & \quad + \left[\frac{\partial \mathbf{R}^1}{\partial \mathbf{X}^0} + \frac{\partial \boldsymbol{R}^1_{\mathrm{GCL}}}{\partial \mathbf{X}^0} \mathbf{Q}^0\right]^T \boldsymbol{\Lambda}_f^1 \end{aligned} \tag{22}$$

The specific form of these equations will be discussed in subsequent sections. With the adjoint coefficients satisfying the flowfield and grid adjoint equations, the sensitivity derivatives are calculated as follows:

$$\begin{aligned} \frac{\mathrm{d}L}{\mathrm{d}\mathbf{D}} = {} & \sum_{n=1}^{N} \left(\frac{\partial f^n}{\partial \mathbf{D}} + [\boldsymbol{\Lambda}_f^n]^T \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{D}} + \frac{\partial \boldsymbol{R}^n_{\mathrm{GCL}}}{\partial \mathbf{D}} \mathbf{Q}^{n-1}\right] + [\boldsymbol{\Lambda}_g^n]^T \frac{\partial \mathbf{G}^n}{\partial \mathbf{D}} \right) \Delta t \\ & + \left(\frac{\partial f^0}{\partial \mathbf{D}} + [\boldsymbol{\Lambda}_f^0]^T \frac{\partial \mathbf{R}^{\mathrm{in}}}{\partial \mathbf{D}} + [\boldsymbol{\Lambda}_g^0]^T \frac{\partial \mathbf{G}^0}{\partial \mathbf{D}} \right) \Delta t \end{aligned} \tag{23}$$

Implementation

Flowfield Adjoint Equations

The implementation and solution of Eqs. (19) and (20) are based largely on the steady-state strategies described in [4,15–19]. In this manner, a great deal of software development effort is avoided because the steady and unsteady equations share many similar terms, namely, the details of the spatial discretization. However, some fundamental differences in the implementation must be addressed for time-dependent problems.

Implications of Reverse Time Integration

Although the discrete solution $\mathbf{Q}^n$ for Eq. (3) is determined by marching forward in physical time from $n = 0$ to N, due to the nature of the adjoint equations and their boundary conditions, the solution for $\boldsymbol{\Lambda}_f^n$ must instead be initiated from $n = N$ and proceed backward in physical time. Because Eqs. (19) and (20) involve the linearizations $\partial \mathbf{R}^n / \partial \mathbf{Q}$ and $\partial f^n / \partial \mathbf{Q}$, the flow solution $\mathbf{Q}^n$ at all time levels must be available during the reverse integration.

In practice, the most straightforward approach to meeting this requirement is to store $\mathbf{Q}^n$ to disk for all n during the solution of Eq. (16). In this case, the storage cost is significant, but the primary advantage is ease of implementation. This is the approach used for the current study. For problems in which the mesh is changing in time, the grid point coordinates and associated speeds are also stored. Although these mesh-related values could be recovered by performing the mesh motion in reverse, ease of the full storage implementation has been favored.

Solution Strategy

As described in [20], each solution vector $\mathbf{Q}^n$ is determined through a dual time-stepping procedure. In this approach, a sequence of subiterations is performed within each physical time step. The procedure relies on an approximate linearization of the discrete residual combined with a pseudotime term to achieve a scheme directly analogous to that used in [22] for steady flows. The same subiterative strategy is employed for the time-dependent adjoint equations, following an approach similar to that outlined in [18]. The Jacobian matrix used to relax the adjoint system is constructed once at each time step n based on the value of $\mathbf{Q}^n$ and does not change during the subiterative procedure.

A requirement for performing adjoint solutions is that the iteration scheme be linearly stable. It has been observed in some cases, more often for unsteady problems than for steady ones, that linear stability is not satisfactory. Suggested explanations [19,26–28] vary from physical instabilities to instabilities of the numerical schemes involved. The generalized conjugate residual scheme described in [29] has been used to wrap the multicolor Gauss–Seidel iteration as well as the temporal subiterative procedure. This approach has been found to work well in stabilizing otherwise problematic iterations.

Data Storage

For three-dimensional dynamic grid simulations using a one-equation turbulence model, the reverse time-integration and solution techniques outlined earlier require the storage of 12 floating-point variables per grid point at each time step: six flowfield variables, three mesh coordinates, and three mesh velocities. For large-scale problems involving many time steps, this strategy can easily result in

a storage requirement on the order of terabytes of disk space. Strategies for circumventing storage limitations have been suggested in the literature [9,30,31]; these may be the focus of future investigations once an initial capability has been established.

In the current implementation, each processor is responsible for reading and writing its local solution for the entire time history to a unique file on disk. Because each file may contain several gigabytes of data, requiring several hundred processors to parse sequential-access files at each time step can be very inefficient. For this reason, direct-access files are used so that the file pointer can be immediately placed at the record of interest. It has been found that this approach can decrease the time required for disk input/output (I/O) by as much as two orders of magnitude for large cases. The use of asynchronous file I/O was also examined, although it is not currently being used.

Grid Adjoint and Sensitivity Equations

Depending on the nature of the grid operator $\mathbf{G}$ and the design variables $\mathbf{D}$, the grid adjoint and sensitivity equations may need to be solved at each time level n, once at $n = 0$, or not at all. If solutions at each time step are required, they are performed at the completion of each step of the adjoint solver, rather than subsequently performing additional loops over the entire range of time levels. In this manner, $\mathbf{Q}^n$, $\mathbf{X}^n$, and the mesh velocities are the only vectors that must be stored for all n, whereas $\mathbf{\Lambda}_f^n$ and $\mathbf{\Lambda}_g^n$ may be discarded when no longer needed.

The predominant challenge in the discretization and solution of Eqs. (21–23) is the infrastructure required to simultaneously manage data from several time levels. An inspection of Eqs. (A7–A9) in the Appendix that are higher-order analogs to Eq. (21) shows that, for a given time step n, the solution for $\mathbf{\Lambda}_g^n$ may depend on values of $\mathbf{Q}$ from adjacent time levels both before and subsequent to level n. Values of $\mathbf{\Lambda}_f$ must also be available at time level n as well as later time levels. Moreover, this complexity increases with the temporal order of the scheme.

The summation term in Eq. (21) is ultimately due to the dependency of the mesh speeds on grid coordinates at multiple time levels, according to the BDF scheme being used. Rather than linearizing $\mathbf{R}$ and $\boldsymbol{R}_{\mathrm{GCL}}$ at several time levels with respect to the grid coordinates at the current time level as indicated in the summation, an inverse approach more amenable to the existing implementation of the spatial linearizations is used. The residual at time level n is linearized with respect to the grid coordinates at every time level in the temporal stencil by seeding the linearizations with the appropriate BDF coefficient. The results are then stored temporarily for use in evaluating the summation term at subsequent time levels within the stencil, after which the linearizations are discarded.

Verification of Adjoint Implementation

To verify the accuracy of the implementation, comparisons are made with results generated through an independent approach based on the use of complex variables. This approach was originally suggested in [32,33] and was first applied to a Navier–Stokes solver in [3]. Using this formulation, an expression for the derivative of a real-valued function $f(x)$ may be found by expanding the function in a complex-valued Taylor series, using an imaginary perturbation $i\varepsilon$:

$$\frac{\partial f}{\partial x} = \frac{\mathrm{Im}[f(x + i\varepsilon)]}{\varepsilon} + \mathcal{O}(\varepsilon^2) \tag{24}$$

The primary advantage of this method is that true second-order accuracy may be obtained by selecting step sizes without concern for subtractive cancellation errors typically present in real-valued divided differences. Through the use of an automated scripting procedure outlined in [34], this capability can be immediately recovered at any time for the baseline flow solver. For computations using this method, the imaginary step size has been chosen to be 10^{-30}, which highlights the robustness of the complex-variable approach. For each verification test, all equations sets are converged to machine precision for both the complex-variable and adjoint approaches. When used, the elasticity matrix $\boldsymbol{K}$ is assumed to be constant throughout the verification.

Static Grid

Test Case

The first test case is used to verify the implementation for unsteady flows on static grids. For this example, fully turbulent flow over the ONERA M6 wing [35] shown in Fig. 1 is considered. The grid contains 16,391 nodes and 90,892 tetrahedral elements, and 16 processors are used for the simulation. The freestream Mach number is 0.3, the angle of attack is 1 deg, and the Reynolds number is 1×10^6 based on the mean aerodynamic chord (MAC). The simulation is initiated from freestream conditions $\mathbf{Q}^\infty$, which leads to $\mathbf{R}^{\mathrm{in}} \equiv \mathbf{Q}^\infty - \mathbf{Q}^0$. The solution is advanced five physical time steps using a nondimensional Δt of 0.1. Although this coarse spatial resolution, relatively large time step, and brief duration of the simulation are not sufficient to resolve the flow physics of the problem, they are adequate to evaluate the discrete consistency of the implementation.

Design Variables

For this test, two general classes of design variables are used. The first class of variables is composed of global parameters unrelated to the computational grid. These variables include parameters such as the freestream Mach number and angle of attack. Such variables are useful in verifying the implementation of the flowfield adjoint equation, as the terms in Eq. (23) associated with these parameters are generally trivial to implement or identically zero, and solution of the mesh adjoint equations is not required.

The second class of design variables provides general shape control of the configuration. The implementation allows the user to employ a geometric parameterization scheme of choice, provided the associated surface grid linearizations are available. For all examples in the current study, the grid parameterization approach described in [36] is used. This approach can be used to define general shape parameterizations of existing grids using a set of aircraft-centric design variables such as camber, thickness, shear, twist, and planform parameters at various locations on the geometry. The user also has the freedom to associate two or more design variables to define more general parameters. In the event that multiple bodies of the same shape are to be designed, the implementation allows for a single set of design variables to be used to simultaneously define such bodies. In this fashion, the shape of each body is constrained to be identical throughout the course of the design.

Grid Adjoint Equation

For this case, there is only one grid operator, $\mathbf{G}(\mathbf{X}, \mathbf{D}) \equiv \mathbf{X}_{\mathrm{surf}} - \boldsymbol{K}\mathbf{X}$, which does not depend on time. As a result, the grid adjoint equation can be recast as

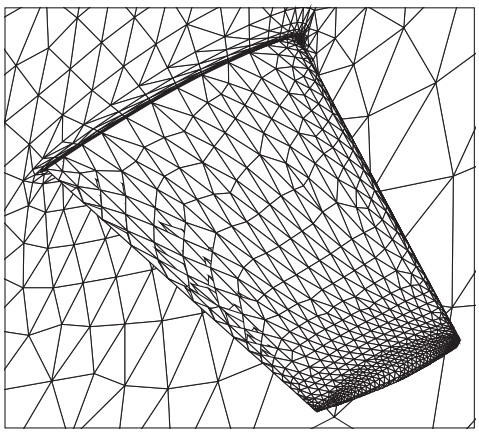


Fig. 1 Surface grid for ONERA M6 wing.

$$\left[\boldsymbol{K}+\frac{\partial \boldsymbol{K}}{\partial \mathbf{X}}\mathbf{X}\right]^T\boldsymbol{\Lambda}_g=\sum_{n=1}^{N}\left\{\left[\frac{\partial f^n}{\partial \mathbf{X}}\right]^T+\left[\frac{\partial \boldsymbol{V}^n}{\partial \mathbf{X}}\frac{\mathbf{Q}^n-\mathbf{Q}^{n-1}}{\Delta t}\right]^T\boldsymbol{\Lambda}_f^n+\sum_{k=0}^{1}\left[\frac{\partial \mathbf{R}^{n+k}}{\partial \mathbf{X}}\right]^T\boldsymbol{\Lambda}_f^{n+k}\right\}+\left[\frac{\partial f^0}{\partial \mathbf{X}}\right]^T+\left[\frac{\partial \mathbf{R}^1}{\partial \mathbf{X}}\right]^T\boldsymbol{\Lambda}_f^1 \tag{25}$$

and the sensitivity derivative is

$$\frac{\mathrm{d}L}{\mathrm{d}\mathbf{D}}=\sum_{n=1}^{N}\left(\frac{\partial f^n}{\partial \mathbf{D}}+[\boldsymbol{\Lambda}_f^n]^T\frac{\partial \mathbf{R}^n}{\partial \mathbf{D}}\right)\Delta t+\left(\frac{\partial f^0}{\partial \mathbf{D}}+[\boldsymbol{\Lambda}_f^0]^T\frac{\partial \mathbf{Q}^\infty}{\partial \mathbf{D}}\right)\Delta t+[\boldsymbol{\Lambda}_g]^T\left[\frac{\partial \mathbf{X}_{\mathrm{surf}}}{\partial \mathbf{D}}-\frac{\partial \boldsymbol{K}}{\partial \mathbf{D}}\mathbf{X}\right]\Delta t \tag{26}$$

Computational Results

The test has been performed using the BDF1 scheme and all other time-integration schemes described in the Appendix, and results are listed in Table 1. Sensitivity derivatives of the lift coefficient at the final time step with respect to the angle of attack and a camber variable located at the midspan of the wing are shown. The results for the adjoint implementation exhibit excellent agreement with the complex-variable approach, differing at most in the fifteenth digit.

Rigidly Moving Grid

Test Case

The next test case is used to verify the implementation for rigidly moving meshes. For this case, the grid and freestream conditions and computational environment are identical to those described for the preceding test; however, the mesh is now subjected to an oscillatory pitch-plunge motion based on the rigid mesh transform approach outlined earlier. The nondimensional pitching and plunging reduced frequencies are 0.5 and 0.1, respectively. The pitching amplitude is 5 deg and takes place about a vector normal to the symmetry plane located 0.47 MAC from the wing root leading edge. The amplitude of the plunging motion is 0.38 MAC. The baseline wing position at $t=0$ is as shown in Fig. 1. As in the preceding test, the simulation is initiated from freestream conditions $\mathbf{R}^{\mathrm{in}}\equiv\mathbf{Q}^\infty-\mathbf{Q}^0$ and is advanced five physical time steps using a nondimensional Δt of 0.1.

Design Variables

The design variables for the current test include those described earlier for the static grid example, as well as a third class of parameters governing the rigid motion procedure described earlier. These include translation and rotation frequencies, amplitudes, and directional vectors, as well as centers of rotation.

Grid Adjoint Equation

For this test case, the following grid operators are used: at the zeroth time level, the grid is either unchanged or governed by the elasticity equations $\mathbf{G}^0(\mathbf{X}^0,\mathbf{D})\equiv\mathbf{X}^0_{\mathrm{surf}}-\mathbf{K}^0\mathbf{X}^0$; grids at other time levels are governed by the rigid motion equation $\mathbf{G}^n(\mathbf{X}^n,\mathbf{D})\equiv\boldsymbol{R}^n\mathbf{X}^0+\boldsymbol{\tau}^n-\mathbf{X}^n$.

The grid adjoint equations are given by

$$\boldsymbol{\Lambda}_g^n=\left[\frac{\partial f}{\partial \mathbf{X}^n}\right]^T+\left[\frac{\partial \boldsymbol{V}^n}{\partial \mathbf{X}^n}\frac{\mathbf{Q}^n-\mathbf{Q}^{n-1}}{\Delta t}\right]^T\boldsymbol{\Lambda}_f^n+\sum_{k=0}^{1}\left[\frac{\partial \mathbf{R}^{n+k}}{\partial \mathbf{X}^n}+\frac{\partial \boldsymbol{R}^{n+k}_{\mathrm{GCL}}}{\partial \mathbf{X}^n}\mathbf{Q}^{n+k-1}\right]^T\boldsymbol{\Lambda}_f^{n+k},\quad \text{for } 1\le n\le N \tag{27}$$

Under the assumption that the shape does not change ($\mathbf{X}^0$ is constant), the sensitivity derivative is given by

$$\frac{\mathrm{d}L}{\mathrm{d}\mathbf{D}}=\sum_{n=1}^{N}\left(\frac{\partial f^n}{\partial \mathbf{D}}+[\boldsymbol{\Lambda}_f^n]^T\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{D}}+\frac{\partial \boldsymbol{R}^n_{\mathrm{GCL}}}{\partial \mathbf{D}}\mathbf{Q}^{n-1}\right]+[\boldsymbol{\Lambda}_g^n]^T\frac{\partial(\boldsymbol{R}^n\mathbf{X}^0+\boldsymbol{\tau}^n)}{\partial \mathbf{D}}\right)\Delta t+\left(\frac{\partial f^0}{\partial \mathbf{D}}+[\boldsymbol{\Lambda}_f^0]^T\frac{\partial \mathbf{Q}^\infty}{\partial \mathbf{D}}\right)\Delta t \tag{28}$$

The formulation that would allow shape design is the following:

$$\left[\boldsymbol{K}^0+\frac{\partial \boldsymbol{K}^0}{\partial \mathbf{X}^0}\mathbf{X}^0\right]^T\boldsymbol{\Lambda}_g^0=\left(\sum_{n=1}^{N}[\boldsymbol{R}^n]^T\boldsymbol{\Lambda}_g^n\right)+\left[\frac{\partial f^0}{\partial \mathbf{X}^0}\right]^T+\left[\frac{\partial \mathbf{Q}^\infty}{\partial \mathbf{X}^0}\right]^T\boldsymbol{\Lambda}_f^0+\left[\frac{\partial \mathbf{R}^1}{\partial \mathbf{X}^0}+\frac{\partial \boldsymbol{R}^1_{\mathrm{GCL}}}{\partial \mathbf{X}^0}\mathbf{Q}^\infty\right]^T\boldsymbol{\Lambda}_f^1 \tag{29}$$

and the corresponding sensitivity derivative is

$$\frac{\mathrm{d}L}{\mathrm{d}\mathbf{D}}=\sum_{n=1}^{N}\left(\frac{\partial f^n}{\partial \mathbf{D}}+[\boldsymbol{\Lambda}_f^n]^T\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{D}}+\frac{\partial \boldsymbol{R}^n_{\mathrm{GCL}}}{\partial \mathbf{D}}\mathbf{Q}^{n-1}\right]+[\boldsymbol{\Lambda}_g^n]^T\frac{\partial \boldsymbol{\tau}^n}{\partial \mathbf{D}}\right)\Delta t+\left(\frac{\partial f^0}{\partial \mathbf{D}}+[\boldsymbol{\Lambda}_f^0]^T\frac{\partial \mathbf{Q}^\infty}{\partial \mathbf{D}}+[\boldsymbol{\Lambda}_g^0]^T\left[\frac{\partial \mathbf{X}^0_{\mathrm{surf}}}{\partial \mathbf{D}}-\frac{\partial \boldsymbol{K}^0}{\partial \mathbf{D}}\mathbf{X}^0\right]\right)\Delta t \tag{30}$$

Computational Results

Results for the derivatives of the lift coefficient at the final time step are shown in Table 2 for the current case. In addition to the angle of attack and camber variables, derivatives with respect to the rigid motion pitching frequency are also shown. The agreement with the complex-variable formulation is excellent for each of the time-integration schemes considered.

Table 1 Results for static grid test case where A denotes adjoint result and C denotes complex-variable result

Design variable	BDF1	BDF2	BDF3	$\mathrm{BDF2}_{\mathrm{opt}}$
Angle of attack	A: 0.004249541855867	A: 0.003734353591935	A: 0.003687377975335	A: 0.003708754474661
	C: 0.004249541855867	C: 0.003734353591935	C: 0.003687377975335	C: 0.003708754474661
Camber	A: 0.010713047647152	A: 0.013701437304586	A: 0.014574974114575	A: 0.014145698047604
	C: 0.010713047647155	C: 0.013701437304586	C: 0.014574974114577	C: 0.014145698047602

Table 2 Results for rigidly moving grid where A denotes adjoint result and C denotes complex-variable result

Design variable	BDF1	BDF2	BDF3	$\mathrm{BDF2}_{\mathrm{opt}}$
Angle of attack	A: 0.004713138571667	A: 0.004293218571759	A: 0.004245785984455	A: 0.004267302756747
	C: 0.004713138571667	C: 0.004293218571759	C: 0.004245785984455	C: 0.004267302756681
Pitching frequency	A: −0.403740396501207	A: −0.527819225717431	A: −0.529833595955533	A: −0.528894917963836
	C: −0.403740396501207	C: −0.527819225717432	C: −0.529833595955533	C: −0.528894917963837
Camber	A: 0.011630821689945	A: 0.013925365539211	A: 0.014291228334440	A: 0.014071544549783
	C: 0.011630821689944	C: 0.013925365539206	C: 0.014291228334428	C: 0.014071544549783

Table 3 Results for morphing grid where A denotes adjoint result and C denotes complex-variable result

Design variable	BDF1	BDF2	BDF3	BDF2_{opt}
Angle of attack	A: 0.004713528355526	A: 0.004298221887378	A: 0.004250753632738	A: 0.004272205860974
	C: 0.004713528355526	C: 0.004298221887378	C: 0.004250753632738	C: 0.004272205860974
Pitching frequency	A: −0.403961428430834	A: −0.528263525075847	A: −0.530205775809711	A: −0.529295291075346
	C: −0.403961428430834	C: −0.528263525075847	C: −0.530205775809710	C: −0.529295291075346
Camber	A: 0.011680362720549	A: 0.013922237526691	A: 0.014268675858452	A: 0.014055458873064
	C: 0.011680362720548	C: 0.013922237526686	C: 0.014268675858435	C: 0.014055458873058

Morphing Grid

Test Case

To evaluate the accuracy of the implementation for morphing grids, the test case used for rigid motion described earlier is repeated with slight modifications. For the current test, the surface grid of the wing is moved using rigid motion, whereas the interior of the mesh is determined using the elasticity relation given by Eq. (9). All other input parameters remain unchanged.

Design Variables

The current test case uses the same design variables as the rigid motion test case described earlier.

Grid Adjoint Equation

At all time levels, the grids are governed by the elasticity equations $\mathbf{G}^n(\mathbf{X}^n, \mathbf{D}) \equiv \mathbf{X}^n_{\text{surf}} - \boldsymbol{K}^0\mathbf{X}^n$, and the surface coordinates are governed by the rigid motion equation $\mathbf{X}^n_{\text{surf}} = \boldsymbol{R}^n\mathbf{X}^0_{\text{surf}} + \boldsymbol{\tau}^n$.

The grid adjoint equations are given by

$$[\boldsymbol{K}^0]^T\boldsymbol{\Lambda}^n_g = \left[\frac{\partial f}{\partial \mathbf{X}^n}\right]^T + \left[\frac{\partial \boldsymbol{V}^n}{\partial \mathbf{X}^n}\frac{\mathbf{Q}^n - \mathbf{Q}^{n-1}}{\Delta t}\right]^T\boldsymbol{\Lambda}^n_f + \sum_{k=0}^{1}\left[\frac{\partial \mathbf{R}^{n+k}}{\partial \mathbf{X}^n} + \frac{\partial \boldsymbol{R}^{n+k}_{\text{GCL}}}{\partial \mathbf{X}^n}\mathbf{Q}^{n+k-1}\right]^T\boldsymbol{\Lambda}^{n+k}_f, \quad \text{for } 1 \le n \le N \tag{31}$$

$$\left[\boldsymbol{K}^0 + \frac{\partial \boldsymbol{K}^0}{\partial \mathbf{X}^0}\mathbf{X}^0\right]^T\boldsymbol{\Lambda}^0_g = \left[\frac{\partial f^0}{\partial \mathbf{X}^0}\right]^T + \left[\frac{\partial \mathbf{Q}^\infty}{\partial \mathbf{X}^0}\right]^T\boldsymbol{\Lambda}^0_f + \left[\frac{\partial \mathbf{R}^1}{\partial \mathbf{X}^0} + \frac{\partial \boldsymbol{R}^1_{\text{GCL}}}{\partial \mathbf{X}^0}\mathbf{Q}^\infty\right]^T\boldsymbol{\Lambda}^1_f - \sum_{n=1}^{N}\left[\frac{\partial \boldsymbol{K}^0}{\partial \mathbf{X}^0}\mathbf{X}^n\right]^T\boldsymbol{\Lambda}^n_g \tag{32}$$

The sensitivity derivative is

$$\frac{\mathrm{d}L}{\mathrm{d}\mathbf{D}} = \sum_{n=1}^{N}\left(\frac{\partial f^n}{\partial \mathbf{D}} + [\boldsymbol{\Lambda}^n_f]^T\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{D}} + \frac{\partial \boldsymbol{R}^n_{\text{GCL}}}{\partial \mathbf{D}}\mathbf{Q}^{n-1}\right] + [\boldsymbol{\Lambda}^n_g]^T\left[\frac{\partial(\boldsymbol{R}^n\mathbf{X}^0_{\text{surf}} + \boldsymbol{\tau}^n)}{\partial \mathbf{D}} - \frac{\partial \boldsymbol{K}^0}{\partial \mathbf{D}}\mathbf{X}^n\right]\right)\Delta t + \left(\frac{\partial f^0}{\partial \mathbf{D}} + [\boldsymbol{\Lambda}^0_f]^T\frac{\partial \mathbf{Q}^\infty}{\partial \mathbf{D}} + [\boldsymbol{\Lambda}^0_g]^T\left[\frac{\partial \mathbf{X}^0_{\text{surf}}}{\partial \mathbf{D}} - \frac{\partial \boldsymbol{K}^0}{\partial \mathbf{D}}\mathbf{X}^0\right]\right)\Delta t \tag{33}$$

Two observations can be made. First, note that in the absence of any surface motion, that is, $\boldsymbol{R}^n$ is the identity matrix and $\boldsymbol{\tau}^n = 0$, the morphing grid formulation is equivalent to the static grid formulation. Also, with a constant transformation matrix T applied to all computational boundaries, the morphing and rigidly moving grid formulations are equivalent.

Computational Results

The results for the current test case are shown in Table 3. Derivatives of the lift coefficient at the final time step with respect to each of the design variables exhibit excellent agreement for the adjoint implementation and complex-variable formulation.

Large-Scale Design Cases

Two large-scale design optimization examples are presented. Although the grid motion in both cases is prescribed, a more realistic treatment would involve the use of additional coupled computational models such as 6 degrees of freedom or structural simulations. Although such capabilities are available for use with the flow solver [20], their effects have not been accounted for in the derivation and implementation of the adjoint equations. This important development is relegated to future work.

Both of the example cases shown next have been performed using 128 dual-socket quad-core nodes with 3.0 GHz Intel Xeon processors in a fully dense fashion for a total of 1024 computational cores. This environment has been chosen to maximize computational efficiency for the chosen test problems; numerical experiments have shown that the solvers used in the current study scale well in this range for the grid sizes selected.

The computational grid sizes and time steps for the examples presented here have been chosen merely to demonstrate optimization capability for typical problems using immediately available resources. Spatial and/or temporal refinement could be readily performed if desired. Although the formulation places no restrictions on initial conditions, all solutions are started from freestream conditions. The grids have been generated using the method in [37], and the optimizations have been performed using a trust region method from the package described in [38].

Tilt-Rotor Configuration

The first large-scale example is a three-bladed tilt-rotor configuration similar to that used by the V-22 aircraft and is based on the tilt-rotor aeroacoustics model (TRAM) geometry described in [39,40]. The grid used for this computation is designed for a blade collective setting of $\Theta = 14$ deg and consists of 5,048,727 nodes and 29,802,252 tetrahedral elements. The rotational speed of the rotor is held constant at a value corresponding to a tip Mach number of 0.62 in a hover condition. The Reynolds number is 2.1×10^6 based on the blade tip chord. The physical time step is chosen to

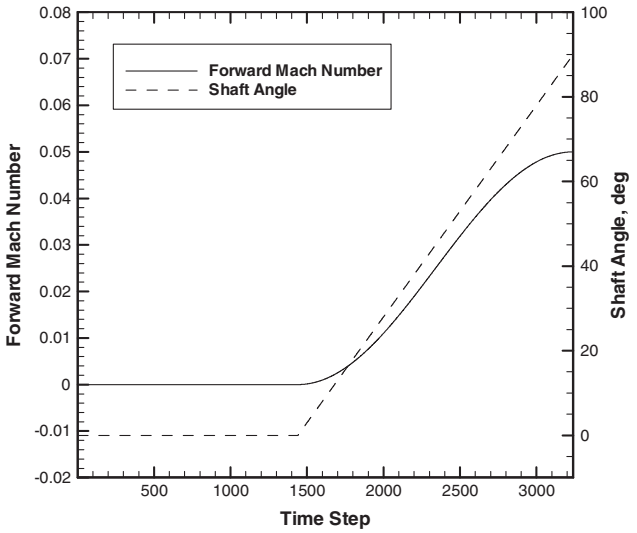


Fig. 2 Forward Mach number and shaft angle schedule for TRAM rotor simulation.

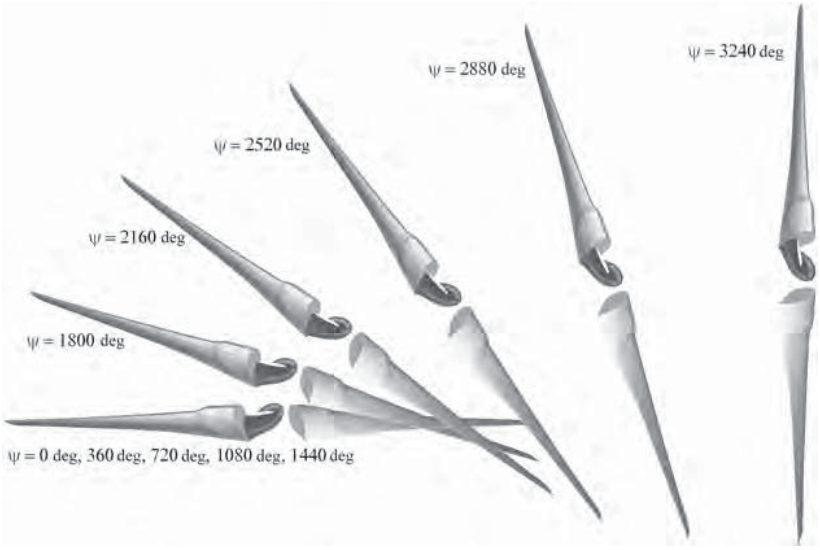


Fig. 3 View of TRAM rotor motion.

correspond to 1 deg of rotor azimuth, for a total of 360 time steps per revolution. The $BDF2_{opt}$ formulation outlined in [41] is used with 10 subiterations per time step.

For this test, the prescribed rigid mesh motion consists of four initial revolutions of the geometry designed to reach a quasi-steady hover condition, followed by five additional revolutions during which a 90 deg constant-rate pitch-up maneuver into a forward-flight mode is performed. A more realistic pitch-up scenario might consist of many more revolutions; however, the prescribed motion was chosen to keep the cost of the computation affordable given the current resources. During the pitch-up phase of the motion, an assumed forward-flight velocity profile based on a simple sine function is imposed through the mesh speed terms. The schedule for the shaft angle and forward-flight velocity is shown in Fig. 2, in which the shaft angle is defined to be 0 deg in the hover condition and 90 deg in forward flight. The resulting motion is shown in Fig. 3, in which a snapshot of the rotor is shown every 360 deg during the course of the motion. An isosurface of the second invariant of the velocity-gradient tensor, also known as the Q criterion from [42], at the time step corresponding to $\Psi = 1440$ deg is shown in Fig. 4. The tip vortex system is maintained for 2–3 revolutions of the rotor.

The objective function for the current test case is to maximize the rotor thrust coefficient over the time interval corresponding to the pitch-up maneuver, $1441 \text{ deg} \leq \Psi \leq 3240 \text{ deg}$:

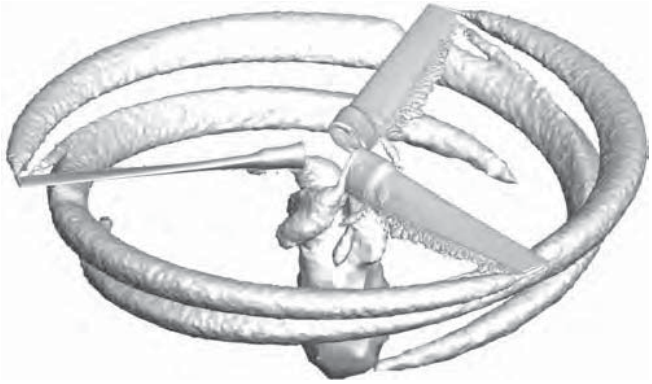


Fig. 4 Isosurface of Q criterion for TRAM rotor at $\Psi = 1440$ deg.

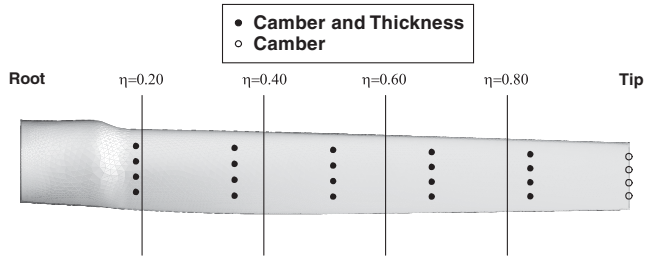


Fig. 6 Spanwise blade and design variable locations for TRAM rotor.

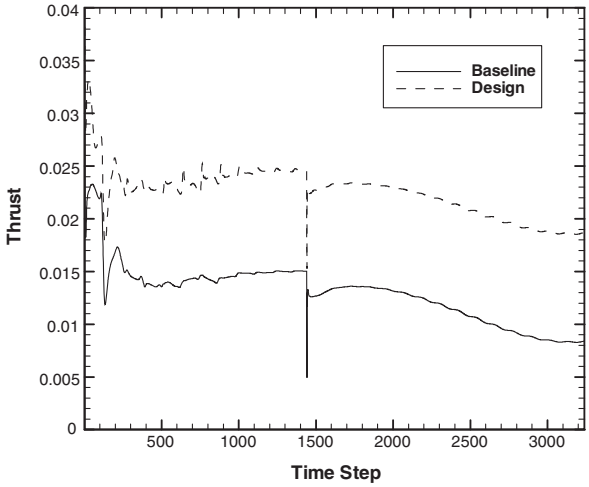


Fig. 5 Thrust for TRAM rotor before and after design optimization.

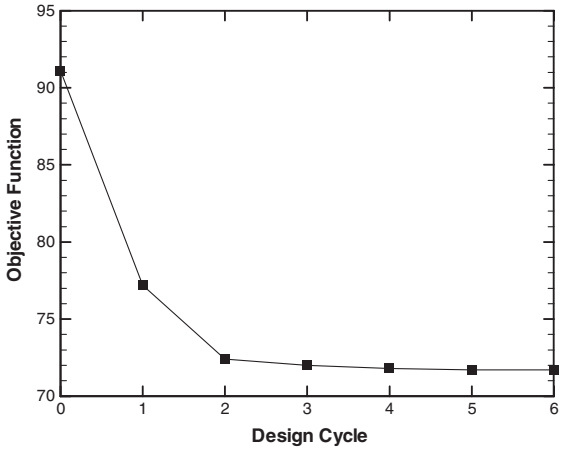


Fig. 7 Objective function history for TRAM rotor.

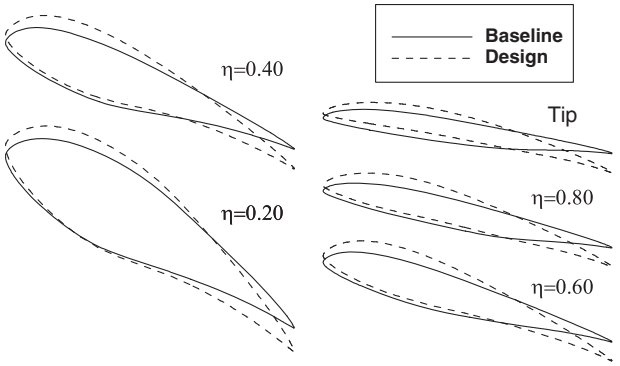


Fig. 8 Spanwise blade cross sections before and after optimization of TRAM rotor.

Fig. 9 Modified F-15 with engine duct geometry.

$$f = \sum_{n=1441}^{3240} (C_T^n - 0.1)^2 \Delta t \tag{34}$$

Here, the target thrust coefficient value of 0.1 has been chosen to sufficiently exceed the baseline thrust profile shown as the solid line in Fig. 5. After the first four rotor revolutions, the thrust coefficient has reached a quasi-steady value of approximately 0.015, which is in good agreement with experimental data given in [39,40]. The thrust coefficient shows a discontinuous behavior at the impulsive start of the pitch-up motion ($n = 1441$) and gradually decreases to a lower constant value in the forward-flight condition. A subtle 3/rev oscillation in the thrust coefficient during the pitch-up maneuver can also be seen.

The surface grid has been parameterized as described in [43]. This approach yields a set of 44 active design variables describing the

Fig. 10 Range of prescribed motion for modified F-15 wing tip.

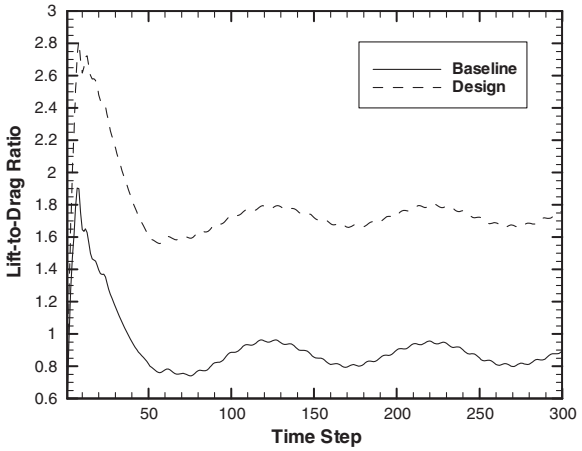


Fig. 11 Lift-to-drag ratio for modified F-15 before and after design optimization.

thickness and camber of the blade geometry as shown in Fig. 6; thinning of the blade is not allowed. Additional bound constraints have been specified based on previous experience in avoiding nonphysical geometries. In addition, a single twist variable is used to modify the blade collective setting during the design.

The convergence history for six design cycles is shown in Fig. 7. The optimizer quickly reduces the value of the objective function over the first two design cycles, after which further improvements are minimal. Closer inspection of the design variables indicates that the majority of values have reached their bound constraints, preventing any further reduction in the objective function. The final thrust coefficient profile is included as the dashed line in Fig. 5. Cross sections of the baseline blade geometry are compared with the optimized geometry in Fig. 8. The optimization has increased the camber of the blade across the span, as well as the blade collective setting.

The cost of each solution to the unsteady flow and adjoint equations for the current example is approximately 3.5 and 10.5 wall-clock hours, respectively; however, due to frequent file I/O, this estimate varies with file system load. The optimization procedure requires 12 calls to the flow solver and 6 calls to the adjoint solver, for a total runtime of approximately 4.5 days of wall-clock time or 110,000 h of CPU time. The disk storage required for one complete flow solution is approximately 1.5 terabytes.

Fighter Jet with Simulated Aeroelastic Effects

The second example uses a deforming grid approach to simulate aeroelastic motion of the modified F-15 fighter jet configuration known as NASA research aircraft 837, shown in Fig. 9.[¶] The computational model assumes half-plane symmetry in the spanwise direction. The grid consists of 4,715,852 nodes and 27,344,343 tetrahedral elements and includes detailed features of the external airframe as well as the internal ducting upstream of the engine fan face and the plenum/nozzle combination downstream of the turbine. For the current test, the freestream Mach number is 0.90, the angle of attack is 0 deg, and the Reynolds number based on the MAC is 1×10^6. The static pressure ratio at the engine fan face is set to 0.9, and the total pressure ratio at the plenum face is ramped linearly from 1.0 to its final value of 5.0 over the first 50 time steps.

The prescribed grid motion consists of 5 Hz 0.3 deg oscillatory rotations of the canard, wing, and tail surfaces about their root chord lines, with the wing oscillations 180 deg out of phase with the canard and tail motion. In addition, the main wing is also subjected to a 5 Hz oscillatory twisting motion for which the amplitude decays linearly from 0.5 deg at the wing tip to 0 deg at the wing root and takes place about the quarter-chord line. This composite motion

[¶]Data available online at http://www.nasa.gov/centers/dryden/aircraft/F-15B-837/index.html [retrieved 4 January 2010].

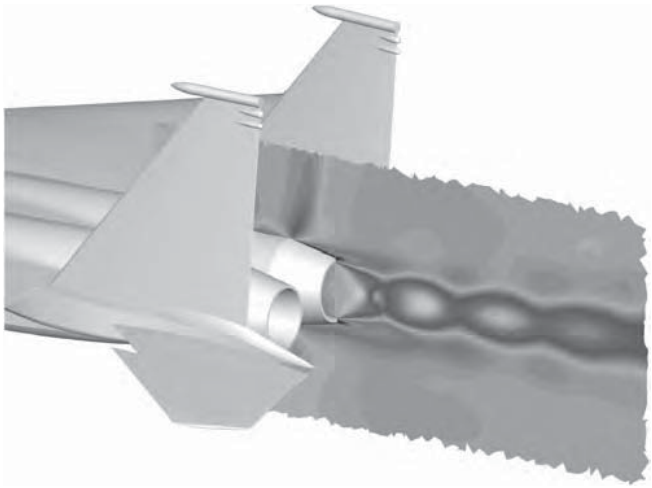


Fig. 12 Cross-section of engine plume contours for modified F-15.

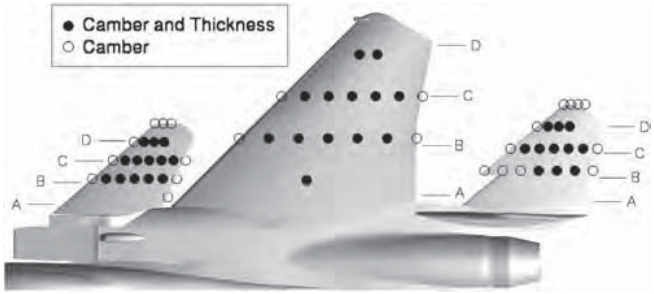


Fig. 13 Spanwise and design variable locations for modified F-15.

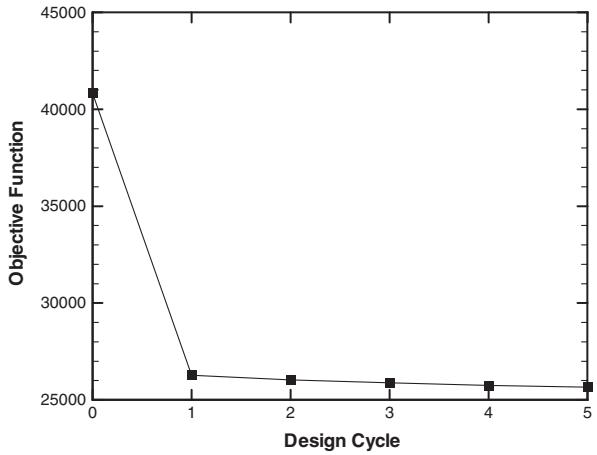


Fig. 14 Objective function history for modified F-15.

results in a maximum wing tip deflection of approximately 1.3% MAC, as shown in Fig. 10. The $BDF2_{opt}$ scheme is used with 10 subiterations and a physical time step corresponding to 100 steps per cycle of grid motion.

The unsteady lift-to-drag ratio (L/D) for the baseline configuration undergoing the specified motion for 300 time steps is shown as the solid line in Fig. 11. The L/D behavior begins to exhibit a periodic response after approximately 100 time steps. The high-frequency oscillations in the profile are believed to be due to a small unsteadiness in the engine plume shown in Fig. 12; this behavior is also present when the mesh is held fixed.

The objective function for the current test case is to maximize L/D for the interval $201 \leq n \leq 300$:

$$f = \sum_{n=201}^{300} [(L/D)^n - 5.0]^2 \Delta t \tag{35}$$

where the target L/D value of 5.0 has been chosen to provide sufficient room for optimization over the baseline profile. The surface grids for the canard, wing, and tail have been parameterized as shown in Fig. 13, resulting in a set of 98 active design variables describing the thickness and camber of each surface. Thinning of the geometry is not permitted, and other bound constraints are chosen to avoid nonphysical geometries.

Convergence of the objective function is shown in Fig. 14. A large reduction in the function is obtained after a single design cycle, after which further improvements are minimal due to many of the design variables having reached their bound constraints. The final L/D profile is included as the dashed line in Fig. 11. The resulting shape changes at various spanwise stations on the canard, wing, and tail are shown in Fig. 15, in which the vertical scale has been exaggerated for clarity. The design procedure has increased the thickness of the wing and canard, as well as the camber across all three elements. Closer inspection shows that the trailing edges of each surface have also been deflected in a downward fashion.

The wall-clock times required for single flow and adjoint solutions for the current problem are approximately 1 and 1.5 h, respectively. For the five design cycles shown in Fig. 14, the optimizer requires 10 flow solutions and 5 adjoint solutions, or a total wall-clock time of approximately 18 h or 18,400 h of CPU time. The disk space necessary to store a single unsteady flow solution is 136 gigabytes.

Conclusions

A discrete adjoint-based methodology for optimization of unsteady flows governed by the three-dimensional Reynolds averaged Navier–Stokes equations on dynamic unstructured grids has been formulated and implemented. The methodology accounts for mesh motion based on both rigid movement as well as deforming grids. The accuracy of the implementation has been verified using comparisons with an independent approach based on the use of complex variables. The methodology has been successfully used in a massively parallel environment to perform two large-scale design optimization examples: one for a tilt rotor in a pitch-up maneuver into a forward-flight regime and another for a fighter jet with simulated aeroelastic effects.

Although the approach outlined in the current study represents significant progress toward the goal of performing routine optimization of unsteady turbulent flows, a number of research areas remain to be explored. The extension of the present formulation to overset grid topologies is ongoing and will allow for the treatment

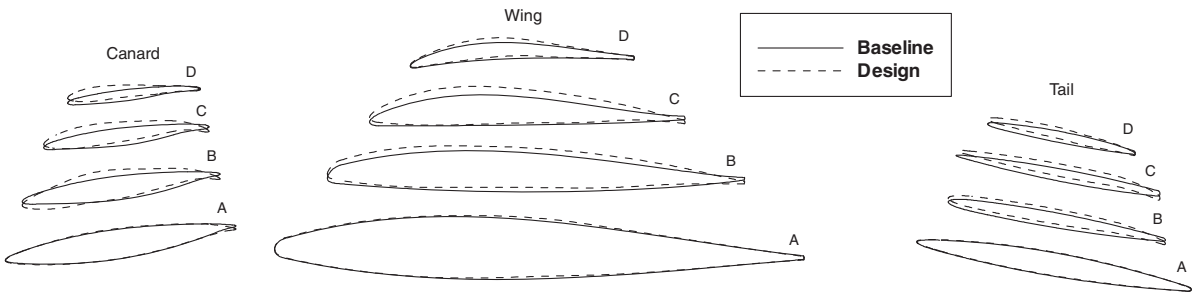


Fig. 15 Canard, wing, and tail cross sections before and after optimization of modified F-15.

of multiple bodies undergoing large relative motion. Methods aimed at reducing the storage costs associated with the flow solution have the potential to drastically reduce disk requirements. Techniques based on variable or adaptive time steps as well as alternate time-integration schemes should be examined. The effects of related computational disciplines such as 6 degrees of freedom and structural models should also be properly accounted for. Finally, the use of the unsteady flowfield adjoint solution holds tremendous potential for performing mathematically rigorous mesh adaptation to specified error bounds.

Appendix A: Adjoint Equations for Higher-Order Backward-Difference-Formula Schemes

The high-order (up to third-order) BDF discretizations for the time derivative of a function s are defined as

$$\frac{\partial s}{\partial t} = \frac{1}{\Delta t}[as^n + bs^{n-1} + cs^{n-2} + ds^{n-3}] \tag{A1}$$

where n is a time level, and the coefficients are given in Table A1. The coefficients listed for the BDF2_{opt} scheme are a linear combination of the BDF2 and BDF3 coefficients taken from [41]. The resulting scheme is second-order-accurate but has a leading truncation error term less than that of the BDF2 scheme. Although usually found to be stable in practice, stability of the BDF2_{opt} and third-order BDF3 scheme are not guaranteed. Discrete conservation laws are defined as

$$aV^n\frac{\mathbf{Q}^n-\mathbf{Q}^{n-1}}{\Delta t} + cV^{n-2}\frac{\mathbf{Q}^{n-2}-\mathbf{Q}^{n-1}}{\Delta t} + dV^{n-3}\frac{\mathbf{Q}^{n-3}-\mathbf{Q}^{n-1}}{\Delta t} + \mathbf{R}^n + \boldsymbol{R}^n_{\text{GCL}}\mathbf{Q}^{n-1} = 0 \tag{A2}$$

Because the morphing grid formulation includes static meshes and rigid motion as special cases, the derivation is provided only for this formulation. Taking into account that $\mathbf{R}^n$ and $\boldsymbol{R}^n_{\text{GCL}}$ are dependent on $\mathbf{X}^{n-2}$ and $\mathbf{X}^{n-3}$, the procedure applied to the BDF1 scheme may also be used to derive the following adjoint equations for the flowfield:

$$\frac{a}{\Delta t}(V^n\Lambda_f^n - V^{n+1}\Lambda_f^{n+1}) + \frac{c}{\Delta t}(V^n\Lambda_f^{n+2} - V^{n-1}\Lambda_f^{n+1}) + \frac{d}{\Delta t}(V^n\Lambda_f^{n+3} - V^{n-2}\Lambda_f^{n+1}) + \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}^n}\right]^T\Lambda_f^n + \boldsymbol{R}^{n+1}_{\text{GCL}}\Lambda_f^{n+1} = -\left[\frac{\partial f^n}{\partial \mathbf{Q}^n}\right]^T, \quad \text{for } 3 \le n \le N \tag{A3}$$

where $\Lambda^{N+1} = \Lambda^{N+2} = \Lambda^{N+3} = 0$; for $n = 2$:

$$\frac{1}{\Delta t}\left(\frac{3}{2}V^2\Lambda_f^2 - aV^3\Lambda_f^3\right) + \frac{c}{\Delta t}(V^2\Lambda_f^4 - V^1\Lambda_f^3) + \frac{d}{\Delta t}(V^2\Lambda_f^5 - V^{\text{in}}\Lambda_f^3) + \left[\frac{\partial \mathbf{R}^2}{\partial \mathbf{Q}^2}\right]^T\Lambda_f^2 + \boldsymbol{R}^3_{\text{GCL}}\Lambda_f^3 = -\left[\frac{\partial f^2}{\partial \mathbf{Q}^2}\right]^T \tag{A4}$$

for $n = 1$:

$$\frac{1}{\Delta t}\left(V^1\Lambda_f^1 - \frac{3}{2}V^2\Lambda_f^2\right) + \frac{1}{\Delta t}\left(cV^1\Lambda_f^3 - \frac{1}{2}V^0\Lambda_f^2\right) + \frac{d}{\Delta t}V^1\Lambda_f^4 + \left[\frac{\partial \mathbf{R}^1}{\partial \mathbf{Q}^1}\right]^T\Lambda_f^1 + \boldsymbol{R}^2_{\text{GCL}}\Lambda_f^2 = -\left[\frac{\partial f^1}{\partial \mathbf{Q}^1}\right]^T \tag{A5}$$

and for the initial conditions:

$$-\frac{1}{\Delta t}V^1\Lambda_f^1 + \frac{1}{2\Delta t}V^0\Lambda_f^2 + \frac{d}{\Delta t}V^0\Lambda_f^3 + \left[\frac{\partial \mathbf{R}^{\text{in}}}{\partial \mathbf{Q}^0}\right]^T\Lambda_f^0 + \boldsymbol{R}^1_{\text{GCL}}\Lambda_f^1 = -\left[\frac{\partial f^0}{\partial \mathbf{Q}^0}\right]^T \tag{A6}$$

The corresponding mesh adjoint equations are obtained as follows. Assuming $\mathbf{R}^{N+1} = \mathbf{R}^{N+2} = \mathbf{R}^{N+3} = 0$ and $\boldsymbol{R}^{N+1}_{\text{GCL}} = \boldsymbol{R}^{N+2}_{\text{GCL}} = \boldsymbol{R}^{N+3}_{\text{GCL}} = 0$:

$$[K^0]^T\Lambda_g^n = \left[\frac{\partial f^n}{\partial \mathbf{X}^n}\right]^T + \left[a\left(\frac{\partial V^n}{\partial \mathbf{X}^n}\frac{\mathbf{Q}^n-\mathbf{Q}^{n-1}}{\Delta t}\right)^T\Lambda_f^n + c\left(\frac{\partial V^n}{\partial \mathbf{X}^n}\frac{\mathbf{Q}^n-\mathbf{Q}^{n+1}}{\Delta t}\right)^T\Lambda_f^{n+2} + d\left(\frac{\partial V^n}{\partial \mathbf{X}^n}\frac{\mathbf{Q}^n-\mathbf{Q}^{n+2}}{\Delta t}\right)^T\Lambda_f^{n+3}\right] + \sum_{k=0}^{3}\left\{\left[\frac{\partial \mathbf{R}^{n+k}}{\partial \mathbf{X}^n} + \frac{\partial \boldsymbol{R}^{n+k}_{\text{GCL}}}{\partial \mathbf{X}^n}\mathbf{Q}^{n+k-1}\right]^T\Lambda_f^{n+k}\right\}, \quad \text{for } 3 \le n \le N \tag{A7}$$

$$[K^0]^T\Lambda_g^2 = \left[\frac{\partial f^2}{\partial \mathbf{X}^2}\right]^T + \left[\frac{3}{2}\left(\frac{\partial V^2}{\partial \mathbf{X}^2}\frac{\mathbf{Q}^2-\mathbf{Q}^1}{\Delta t}\right)^T\Lambda_f^2 + c\left(\frac{\partial V^2}{\partial \mathbf{X}^2}\frac{\mathbf{Q}^2-\mathbf{Q}^3}{\Delta t}\right)^T\Lambda_f^4 + d\left(\frac{\partial V^2}{\partial \mathbf{X}^2}\frac{\mathbf{Q}^2-\mathbf{Q}^4}{\Delta t}\right)^T\Lambda_f^5\right] + \sum_{k=0}^{3}\left\{\left[\frac{\partial \mathbf{R}^{2+k}}{\partial \mathbf{X}^2} + \frac{\partial \boldsymbol{R}^{2+k}_{\text{GCL}}}{\partial \mathbf{X}^2}\mathbf{Q}^{1+k}\right]^T\Lambda_f^{2+k}\right\}, \quad \text{for } n = 2 \tag{A8}$$

$$[K^0]^T\Lambda_g^1 = \left[\frac{\partial f^1}{\partial \mathbf{X}^1}\right]^T + \left[\left(\frac{\partial V^1}{\partial \mathbf{X}^1}\frac{\mathbf{Q}^1-\mathbf{Q}^0}{\Delta t}\right)^T\Lambda_f^1 + c\left(\frac{\partial V^1}{\partial \mathbf{X}^1}\frac{\mathbf{Q}^1-\mathbf{Q}^2}{\Delta t}\right)^T\Lambda_f^3 + d\left(\frac{\partial V^1}{\partial \mathbf{X}^1}\frac{\mathbf{Q}^1-\mathbf{Q}^3}{\Delta t}\right)^T\Lambda_f^4\right] + \sum_{k=0}^{3}\left\{\left[\frac{\partial \mathbf{R}^{1+k}}{\partial \mathbf{X}^1} + \frac{\partial \boldsymbol{R}^{1+k}_{\text{GCL}}}{\partial \mathbf{X}^1}\mathbf{Q}^k\right]^T\Lambda_f^{1+k}\right\}, \quad \text{for } n = 1 \tag{A9}$$

and for the initial conditions, $\mathbf{R}^{\text{in}} \equiv \mathbf{Q}^\infty - \mathbf{Q}^0$:

$$\left[K^0 + \frac{\partial \mathbf{K}^0}{\partial \mathbf{X}^0}\mathbf{X}^0\right]^T\Lambda_g^0 = \left[\frac{\partial f^0}{\partial \mathbf{X}^0}\right]^T + \left[\frac{1}{2}\left(\frac{\partial V^0}{\partial \mathbf{X}^0}\frac{\mathbf{Q}^0-\mathbf{Q}^1}{\Delta t}\right)^T\Lambda_f^2 + d\left(\frac{\partial V^0}{\partial \mathbf{X}^0}\frac{\mathbf{Q}^0-\mathbf{Q}^2}{\Delta t}\right)^T\Lambda_f^3\right] + \left[\frac{\partial \mathbf{Q}^\infty}{\partial \mathbf{X}^0}\right]^T\Lambda_f^0 + \sum_{k=1}^{3}\left\{\left[\frac{\partial \mathbf{R}^k}{\partial \mathbf{X}^0} + \frac{\partial \boldsymbol{R}^k_{\text{GCL}}}{\partial \mathbf{X}^0}\mathbf{Q}^{k-1}\right]^T\Lambda_f^k\right\} - \sum_{n=1}^{N}\left[\frac{\partial \mathbf{K}^0}{\partial \mathbf{X}^0}\mathbf{X}^n\right]^T\Lambda_g^n \tag{A10}$$

The sensitivity derivative for the higher-order BDF schemes is evaluated using Eq. (23).

Table A1 Coefficients for higher-order BDF schemes

Scheme	a	b	c	d
BDF2	3/2	−2	1/2	0
BDF3	11/6	−3	3/2	−1/3
BDF2_{opt}	5.08/3	−2.58	1.08	−0.58/3

Acknowledgments

The authors wish to thank Robert Biedron of NASA Langley Research Center for many useful discussions pertaining to the current work. Geometric parameterizations provided by Bill Jones of NASA Langley Research Center are also appreciated. Jan-Renee Carlson of NASA Langley Research Center is acknowledged for her assistance with plenum geometry modifications and boundary conditions for the fighter jet example. The second and third authors acknowledge the support from NASA under grant NNL07AA23C.

References

[1] Vanderplaats, G. N., Hicks, R. N., and Murman, E. M., "Application of Numerical Optimization Techniques to Airfoil Design," *NASA Conference on Aerodynamic Analysis Requiring Advanced Computers*, NASA SP-347, Part 2, March 1975.

[2] Baysal, O., and Eleshaky, M. E., "Aerodynamic Sensitivity Analysis Methods for the Compressible Euler Equations," *Journal of Fluids Engineering*, Vol. 113, 1991, pp. 681–688.

doi:10.1115/1.2926534
[3] Anderson, W. K., Newman, J. C., Whitfield, D. L., and Nielsen, E. J., "Sensitivity Analysis for the Navier–Stokes Equations on Unstructured Meshes Using Complex Variables," *AIAA Journal*, Vol. 39, No. 1, 2001, pp. 56–63. doi:10.2514/2.1270
[4] Nielsen, E. J., and Park, M. A., "Using an Adjoint Approach to Eliminate Mesh Sensitivities in Computational Design," *AIAA Journal*, Vol. 44, No. 5, 2006, pp. 948–953. doi:10.2514/1.16052
[5] Newman, J. C. III, Taylor, A. C. III, Barnwell, R. W., Newman, P. A., and Hou, G. J.-W., "Overview of Sensitivity Analysis and Shape Optimization for Complex Aerodynamic Configurations," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 87–96. doi:10.2514/2.2416
[6] Fidkowski, K. J., and Darmofal, D. L., "Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics: Overview and Recent Results," AIAA Paper 2009-1303, 2009.
[7] Nadarajah, S., and Jameson, A., "Optimal Control of Unsteady Flows Using Time Accurate and Non-Linear Frequency Domain Methods," AIAA Paper 2002-5436, 2002.
[8] Choi, S., Potsdam, M., Lee, K., Iaccarino, G., and Alonso, J. J., "Helicopter Rotor Design Using a Time-Spectral and Adjoint-Based Method," AIAA Paper 2008-5810, 2008.
[9] Gunzburger, M. D., *Perspectives in Flow Control and Optimization*, Society for Industrial and Applied Mathematics, Philadelphia, 2003.
[10] Mavriplis, D. J., "Solution of the Unsteady Discrete Adjoint for Three-Dimensional Problems on Dynamically Deforming Unstructured Meshes," AIAA Paper 2008-727, 2008.
[11] Mani, K., and Mavriplis, D. J., "Unsteady Discrete Adjoint Formulation for Two-Dimensional Flow Problems with Deforming Meshes," *AIAA Journal*, Vol. 46, No. 6, 2008, pp. 1351–1364. doi:10.2514/1.29924
[12] Muldoon, F., "Control of a Simplified Unsteady Film-Cooling Flow Using Gradient-Based Optimization," *AIAA Journal*, Vol. 46, No. 10, 2008, pp. 2443–2458. doi:10.2514/1.34120
[13] Rumpfkeil, M. P., and Zingg, D. W., "A General Framework for the Optimal Control of Unsteady Flows with Applications," AIAA Paper 2007-1128, 2007.
[14] Yamaleev, N. K., Diskin, B., and Nielsen, E. J., "Adjoint-Based Methodology for Time-Dependent Optimization," AIAA Paper 2008-5857, 2008.
[15] Nielsen, E. J., "Aerodynamic Design Sensitivities on an Unstructured Mesh Using the Navier–Stokes Equations and a Discrete Adjoint Formulation," Ph.D. Dissertation, Dept. of Aerospace and Ocean Engineering, Virginia Polytechnic Inst. and State Univ., Dec. 1998.
[16] Nielsen, E. J., and Anderson, W. K., "Aerodynamic Design Optimization on Unstructured Meshes Using the Navier–Stokes Equations," *AIAA Journal*, Vol. 37, No. 11, 1999, pp. 1411–1419. doi:10.2514/2.640
[17] Nielsen, E. J., and Anderson, W. K., "Recent Improvements in Aerodynamic Design Optimization on Unstructured Meshes," *AIAA Journal*, Vol. 40, No. 6, 2002, pp. 1155–1163. doi:10.2514/2.1765
[18] Nielsen, E. J., Lu, J., Park, M. A., and Darmofal, D. L., "An Implicit, Exact Dual Adjoint Solution Method for Turbulent Flows on Unstructured Grids," *Computers and Fluids*, Vol. 33, No. 9, 2004, pp. 1131–1155. doi:10.1016/j.compfluid.2003.09.005
[19] Nielsen, E. J., and Kleb, W. L., "Efficient Construction of Discrete Adjoint Operators on Unstructured Grids by Using Complex Variables," *AIAA Journal*, Vol. 44, No. 4, 2006, pp. 827–836. doi:10.2514/1.15830
[20] Biedron, R. T., and Thomas, J. L., "Recent Enhancements to the FUN3D Flow Solver for Moving Mesh Applications," AIAA Paper 2009-1360, 2009.
[21] Thomas, P. D., and Lombard, C. K., "Geometrical Conservation Law and Its Application to Flow Computations on Moving Grids," *AIAA Journal*, Vol. 17, No. 10, 1979, pp. 1030–1037. doi:10.2514/3.61273
[22] Anderson, W. K., and Bonhaus, D. L., "An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids," *Computers and Fluids*, Vol. 23, No. 1, 1994, pp. 1–21. doi:10.1016/0045-7930(94)90023-X
[23] Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357–372. doi:10.1016/0021-9991(81)90128-5
[24] Spalart, P. R., and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA Paper 1992-439, 1992.
[25] Saad, Y., and Schultz, M. H., "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No. 3, 1986, pp. 856–869. doi:10.1137/0907058
[26] Elliott, J., "Aerodynamic Optimization Based on the Euler and Navier–Stokes Equations Using Unstructured Grids," Ph.D. Dissertation, Dept. of Aeronautics and Astronautics, Massachusetts Inst. of Technology, Cambridge, MA, June 1998.
[27] Campobasso, M., and Giles, M., "Effects of Flow Instabilities on the Linear Analysis of Turbomachinery Aeroelasticity," *Journal of Propulsion and Power*, Vol. 19, No. 2, 2003, pp. 250–259. doi:10.2514/2.6106
[28] Krakos, J., and Darmofal, D. L., "Effect of Small-Scale Unsteadiness on Adjoint-Based Output Sensitivity," AIAA Paper 2009-4274, 2009.
[29] Eisenstat, S. C., Elman, H. C., and Schultz, M., "Variational Iterative Methods for Nonsymmetric Systems of Linear Equations," *SIAM Journal on Numerical Analysis*, Vol. 20, No. 2, 1983, pp. 345–357. doi:10.1137/0720023
[30] Hinze, M., Walther, A., and Sternberg, J., "Discrete Approximation Schemes for Reduced Gradients and Reduced Hessians in Navier–Stokes Control Utilizing an Optimal Memory-Reduced Procedure for Calculating Adjoints," *Optimal Control Applications and Methods*, Vol. 27, No. 1, 2006, pp. 19–40. doi:10.1002/oca.771
[31] Yamaleev, N., Diskin, B., and Nielsen, E., "Local-in-Time Adjoint-Based Method for Design Optimization of Unsteady Compressible Flows," AIAA Paper 2009-1169, January 2009.
[32] Lyness, J. N., "Numerical Algorithms Based on the Theory of Complex Variables," *Proceedings of the ACM 22nd National Conference*, Thomas Book Co., Washington, DC, 1967, pp. 124–134.
[33] Lyness, J. N., and Moler, C. B., "Numerical Differentiation of Analytic Functions," *SIAM Journal on Numerical Analysis*, Vol. 4, 1967, pp. 202–210. doi:10.1137/0704019
[34] Kleb, W. L., Nielsen, E. J., Gnoffo, P. A., Park, M. A., and Wood, W. A., "Collaborative Software Development in Support of Fast Adaptive Aerospace Tools (FAAST)," AIAA Paper 2003-3978, 2003.
[35] Schmitt, V., and Charpin, F., "Pressure Distributions on the ONERA M6 Wing at Transonic Mach Numbers," *Experimental Database for Computer Program Assessment*, AGARD AR-138, May 1979, pp. B1-1–B1-44.
[36] Samareh, J. A., "A Novel Shape Parameterization Approach," NASA TM-1999-209116, May 1999.
[37] Pirzadeh, S., "Three-Dimensional Unstructured Viscous Grids by the Advancing Front Method," *AIAA Journal*, Vol. 34, No. 1, 1996, pp. 43–49. doi:10.2514/3.13019
[38] Kaufman, L., and Gay, D., "PORT Library: Optimization and Mathematical Programming—User's Manual," Bell Lab., Paris, 1997.
[39] Young, L. A., Booth, E. R., Jr., Yamauchi, G. K., Botha, G., and Dawson, S., "Overview of the Testing of a Small-Scale Proprotor," *American Helicopter Society 55th Annual Forum*, American Helicopter Society, Alexandria, VA, May 1999.
[40] Swanson, S. M., McCluer, M. S., Yamauchi, G. K., and Swanson, A. A., "Airloads Measurements from a $\frac{1}{4}$-Scale Tiltrotor Wind Tunnel Test," *25th European Rotorcraft Forum*, Sept. 1999.
[41] Nyukhtikov, M., Smelova, N., Mitchell, B. E., and Holmes, D. G., "Optimized Dual-Time Stepping Technique for Time-Accurate Navier–Stokes Calculation," *Proceedings of the 10th International Symposium on Unsteady Aerodynamics, Aeroacoustics, and Aeroelasticity of Turbomachines*, Springer, Dordrecht, The Netherlands, 2003, pp. 449–462.
[42] Jeong, J., and Hussain, F., "On the Identification of a Vortex," *Journal of Fluid Mechanics*," Vol. 285, 1995, pp. 69–94. doi:10.1017/S0022112095000462
[43] Nielsen, E. J., Lee-Rausch, E. M., and Jones, W. T., "Adjoint-Based Design of Rotors Using the Navier–Stokes Equations in a Noninertial Reference Frame," *American Helicopter Society 65th Annual Forum*, American Helicopter Society, Alexandria, VA, May 2009.

Z. Wang
Associate Editor

Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.

This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/copyright

Author's personal copy

Computers & Fluids 41 (2011) 82–93

Contents lists available at ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

A critical study of agglomerated multigrid methods for diffusion on highly-stretched grids

James L. Thomas [a,*], Boris Diskin [b,1,2], Hiroaki Nishikawa [b,3]

[a] NASA Langley Research Center, Computational AeroSciences Branch, Mail Stop 128, Hampton, VA 23681, United States
[b] National Institute of Aerospace, 100 Exploration Way, Hampton, VA 23666, United States

ARTICLE INFO

Article history:
Available online 21 September 2010

Keywords:
Multigrid
Agglomeration
Unstructured
Diffusion
Analysis

ABSTRACT

Agglomerated multigrid methods for unstructured grids are studied critically for solving a model diffusion equation on highly-stretched grids typical of practical viscous simulations, following a previous work focused on isotropic grids. Different primal elements, including prismatic and tetrahedral elements in three dimensions, are considered. The components of an efficient node-centered full-coarsening multigrid scheme are identified and assessed using quantitative analysis methods. Fast grid-independent convergence is demonstrated for mixed-element grids composed of tetrahedral elements in the isotropic regions and prismatic elements in the highly-stretched regions. Implicit lines natural to advancing-layer/advancing-front grid generation techniques are essential elements of both relaxation and agglomeration. On agglomerated grids, consistent average-least-square discretizations augmented with edge-directional gradients to increase *h*-ellipticity of the operator are used. Simpler (edge-terms-only) coarse-grid discretizations are also studied and shown to produce grid-dependent convergence—only effective on grids with minimal skewing.

Published by Elsevier Ltd.

1. Introduction

Multigrid techniques [18] are routinely used to accelerate convergence of Reynolds-Averaged Navier–Stokes solvers for large-scale steady and unsteady flow applications, especially within structured-grid methods. Agglomerated multigrid methods for large-scale unstructured-grid applications have also been developed and demonstrated impressive improvements in efficiency over single-grid computations [9–12]. The performance of multigrid solvers is as yet far from the textbook multigrid efficiency goal—converging algebraic errors below discretization errors in the work equivalent to a few residual evaluations; such performance has only been demonstrated to date for relatively simple applications [15,16]. Design of efficient multigrid solvers for unstructured-grid applications is significantly more challenging because analysis tools to understand and predict multigrid performance are less developed than tools for structured grids. In particular, local Fourier analysis (LFA) is widely used on structured grids but is inapplicable to irregular grids. The quantitative analysis tools, idealized relaxation and idealized coarse grid, developed earlier [2] are applicable. These tools, in combination with windowing techniques [3,17], isolate the sources of difficulties and are proving useful to improve both accuracy and efficiency in an unstructured-grid setting.

One of the key weaknesses identified by Venkatakrishnan [19] for unstructured agglomeration methods was the coarse-grid discretization of diffusion (viscous terms). The current approaches for the coarse-grid discretization of diffusion were critically studied for two- and three-dimensional isotropic grids in a previous paper [13]. Direct-discretization and Galerkin approaches were investigated for a model problem representative of laminar diffusion in the incompressible limit. Consistency of coarse-grid discretization was found to be essential for attaining fast grid-independent convergence; consistent discretizatons on agglomerated grids were obtained through direct discretization with an average-least-square approach. Multigrid with coarse grids discretized using either a Galerkin approach or an approximate edge-terms-only direct discretization was also studied but, with both of these approaches, the convergence depended on the grid (particularly skewness) and deteriorated on finer grids. In this paper, we address higher aspect ratios and highly-stretched three-dimensional grids and use only direct discretizations.

* Corresponding author.
E-mail addresses: James.L.Thomas@nasa.gov (J.L. Thomas), bdiskin@nianet.org (B. Diskin), hiro@nianet.org (H. Nishikawa).

[1] This author was supported by the National Institute of Aerospace under the NASA Fundamental Aeronautics Program through NASA Contracts NNL07AA23C and NNL07AA31C.

[2] University of Virginia, Department of Mechanical and Aerospace Engineering Department, 122 Engineer's Way, Charlottesville, Virginia 22904-4746, United States.

[3] This author was supported by the National Institute of Aerospace under the NASA Fundamental Aeronautics Program through NASA Contract NNL07AA23C.

0045-7930/$ - see front matter Published by Elsevier Ltd.
doi:10.1016/j.compfluid.2010.09.023

Author's personal copy

Many applications use grids generated with advancing-layer/advancing-front techniques in which the grids are highly stretched predominantly in the direction normal to the boundary. In this paper, highly-stretched grids transitioning to isotropic grids are considered. The isotropic grids are irregular tetrahedral grids. The highly-stretched grids are mixed-element grids, composed of prismatic and tetrahedral elements; the prismatic grids extend from the surface, where the aspect ratio is highest, to locations where the aspect ratio approaches unity. A full-coarsening/line-implicit multigrid is pursued herein. The coarsening strategy is similar to that used by Hyams et al. [8], although the coarse-grid discretizations are quite different. In [8], a Galerkin coarse-grid construction that is inconsistent for diffusion was used; a direct discretization on the coarse grid was also used but no details of the treatment of viscous terms are given. Mavriplis [9–12] used a directional-coarsening strategy—coarsening by a factor of four in the direction normal to the boundary within the highly-stretched (viscous) regions of the grid; a full coarsening strategy was used in the isotropic (inviscid) regions of the grid. The coarse-grid discretization of viscous terms was through an edge-terms-only direct discretization or a heuristically-scaled Galerkin formulation.

This paper is organized as follows. The discretization schemes for the model diffusion equation are presented in Section 2 from a general finite-volume discretization standpoint. Element-based and element-free schemes are shown; the latter includes certain edge-based discretizations and discretizations on agglomerated grids. The grid agglomeration techniques are presented in Section 3 and Appendix A. The multigrid algorithm, including relaxation and residual-averaging techniques, is described in Section 4. The key ingredients enabling successful multigrid performance are identified and assessed using quantitative analysis methods in Section 5 and Appendices B–D. Three-dimensional multigrid computations demonstrating grid-independent convergence for both isotropic and highly-stretched grids within an ellipsoidal domain are shown in Section 6. The final Section 7 contains conclusions.

2. Discretization schemes

The considered model problem is the Poisson equation

$$\Delta U = f, \tag{1}$$

subject to Dirichlet boundary conditions; function f is a forcing function. The finite-volume discretization (FVD) schemes are derived from the integral form of a conservation law

$$\oint_{\partial\Omega} \nabla U \cdot \hat{\mathbf{n}}\, ds = \int_{\Omega} f\, d\Omega, \tag{2}$$

where ∇U is the solution gradient, Ω is a control volume with boundary $\partial\Omega$, and $\hat{\mathbf{n}}$ is the outward unit normal vector. The general FVD approach requires partitioning the domain into a set of non-overlapping control volumes and numerically implementing Eq. (2) over each control volume.

Node-centered discretizations are considered in which the solutions are defined at the mesh nodes. The discrete schemes described below are representative of viscous discretizations used in Reynolds-Averaged Navier–Stokes unstructured-grid codes. Dirichlet boundary conditions are implemented strongly.

2.1. Element-based discretizations

The target meshes are compositions of primal elements (cells)—triangular and quadrilateral elements in two dimensions (2D) and tetrahedral, hexahedral, prismatic, and pyramidal elements in three dimensions (3D). Control volumes are constructed around the mesh nodes by the median-dual partition (Fig. 1) [1,7].

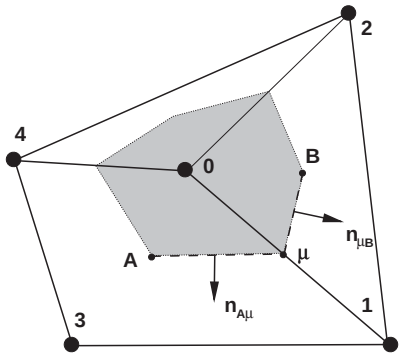


Fig. 1. Illustration for gradient construction; dual volume is shaded.

The target discretization is the Green–Gauss scheme [4]—widely used in node-centered codes and equivalent to a Galerkin finite-element (linear-element) discretization for triangular/tetrahedral grids. For mixed elements, edge derivatives are used to increase the h-ellipticity [18] of the diffusion operator [4,7] and, thus, avoid checkerboard instabilities. It has been shown [3,17] that the scheme possesses second-order accuracy for viscous fluxes on general mixed-element grids.

With reference to Fig. 1 illustrating a mixed-element 2D grid, the scheme approximates the integral flux through the dual faces adjacent to the edge [0,1] as

$$\int_{A\mu B} \nabla U \cdot \hat{\mathbf{n}}\, ds = \nabla U_{A\mu} \cdot \mathbf{n}_{A\mu} + \nabla U_{\mu B} \cdot \mathbf{n}_{\mu B}, \tag{3}$$

where μ is the median of the edge [0,1], subscripts designate dual faces, and $\mathbf{n}_{A\mu}$ and $\mathbf{n}_{\mu B}$ are directed-area vectors. The gradient is reconstructed separately at each dual face as follows. For the triangular element contribution, the gradient is determined from a Green–Gauss evaluation at the primal element,

$$\nabla U_{\mu B} = \overline{\nabla U}_{012}. \tag{4}$$

The gradient overbar denotes a gradient evaluated by the Green–Gauss formula on the primal cell identified by the point subscripts. For the quadrilateral element contribution, the gradient $\nabla U_{A\mu}$ is formed by augmenting the Green–Gauss gradient within the element, $\overline{\nabla U}_{0134}$, with the directional derivative along the edge, $\partial^e U$, defined as

$$\partial^e U = (U_1 - U_0)/|\mathbf{r}_1 - \mathbf{r}_0|, \tag{5}$$

where U_i and $\mathbf{r}_i$ are the solution and the coordinate vector of the node i.

Two approaches to construct the augmented gradient $\nabla U_{A\mu}$ have been used and are illustrated in Fig. 2 for 2D. To present the approaches, the unit vector aligned with the edge [0,1] is defined as

$$\hat{\mathbf{e}} = (\mathbf{r}_1 - \mathbf{r}_0)/|\mathbf{r}_1 - \mathbf{r}_0|, \tag{6}$$

the unit vector normal to the control-volume face is defined as

$$\hat{\mathbf{n}} = \mathbf{n}_{A\mu}/|\mathbf{n}_{A\mu}|, \tag{7}$$

and the Green–Gauss gradient is defined as

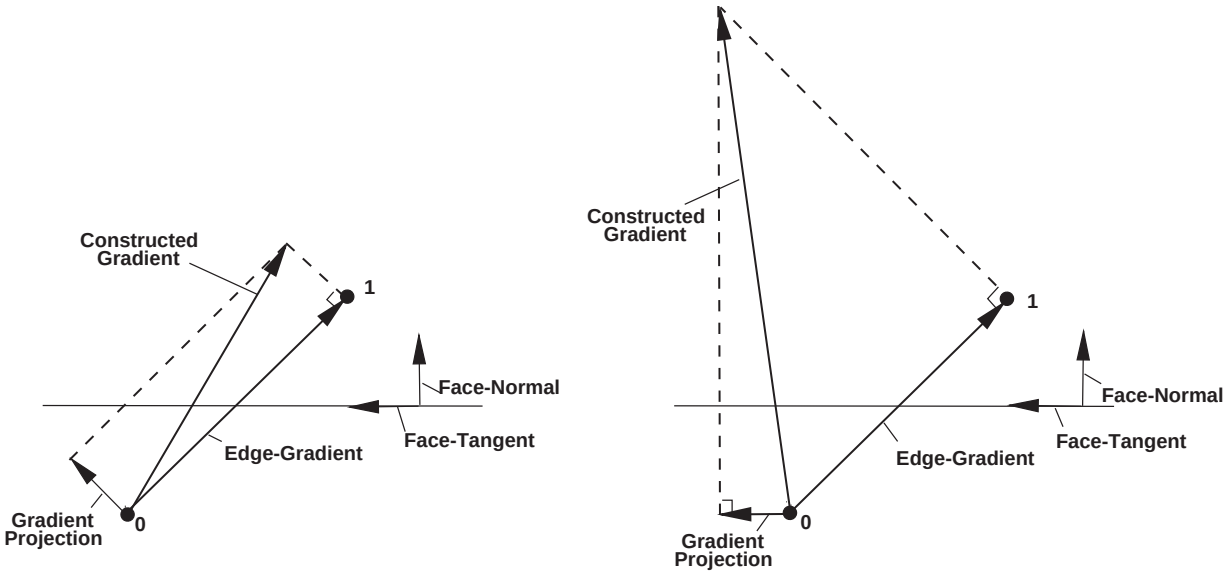
$$\overline{\nabla U} = \overline{\nabla U}_{0134}. \tag{8}$$

The face skew angle θ is the angle between the edge direction and the face-normal direction,

$$\cos\theta = \hat{\mathbf{e}} \cdot \hat{\mathbf{n}}. \tag{9}$$

The first augmentation, probably more widely used and designated here as *edge-normal* (EN), is illustrated in Fig. 2a and enforces that the constructed gradient, $\nabla U^{EN}_{A\mu}$, recovers

Author's personal copy



(a) Edge-normal construction; gradient projection is $\overline{\nabla U} - (\overline{\nabla U} \cdot \hat{\mathbf{e}})\hat{\mathbf{e}}$.

(b) Face-tangent construction; gradient projection is $(\overline{\nabla U} \cdot \hat{\mathbf{f}})\hat{\mathbf{f}}$.

Fig. 2. Illustration of gradient constructions at a control-volume face separating nodes 0 and 1; $\theta = \pi/4$; the edge gradient has magnitude $\partial^e U$ and is oriented in the $\hat{\mathbf{e}}$ direction.

(1) the edge-directional gradient, $\partial^e U\hat{\mathbf{e}}$ and
(2) the Green–Gauss gradient projected onto the plane normal to $\hat{\mathbf{e}}$,

$$\nabla U_{A\mu}^{EN} = (\partial^e U - (\overline{\nabla U} \cdot \hat{\mathbf{e}}))\hat{\mathbf{e}} + \overline{\nabla U}. \tag{10}$$

The second augmentation, designated as *face-tangent* (FT), is illustrated in Fig. 2b and enforces that the constructed gradient, $\nabla U_{A\mu}^{FT}$, recovers

(1) the edge-directional gradient and
(2) the Green–Gauss gradient projected onto the plane normal to $\hat{\mathbf{n}}$,

$$\nabla U_{A\mu}^{FT} = \frac{1}{\hat{\mathbf{n}} \cdot \hat{\mathbf{e}}} \partial^e U\hat{\mathbf{n}} + (\overline{\nabla U} \cdot \hat{\mathbf{f}}) \left[\hat{\mathbf{f}} - \frac{\hat{\mathbf{f}} \cdot \hat{\mathbf{e}}}{\hat{\mathbf{n}} \cdot \hat{\mathbf{e}}}\hat{\mathbf{n}}\right], \tag{11}$$

where $\hat{\mathbf{f}}$ is a unit vector normal to $\hat{\mathbf{n}}$. Note that (11) applies only to 2D but there is an obvious 3D counterpart. The corresponding contributions to the diffusion operator (for the orientation shown in Fig. 2) are given below:

$$\nabla U_{A\mu}^{EN} \cdot \mathbf{n}_{A\mu} = |\mathbf{n}_{A\mu}|[\cos\theta(\partial^e U - (\overline{\nabla U} \cdot \hat{\mathbf{e}})) + \overline{\nabla U} \cdot \hat{\mathbf{n}}], \tag{12}$$

$$\nabla U_{A\mu}^{FT} \cdot \mathbf{n}_{A\mu} = |\mathbf{n}_{A\mu}| \frac{1}{\cos\theta}[\partial^e U + (\overline{\nabla U} \cdot \hat{\mathbf{f}}) \sin\theta]. \tag{13}$$

Both approaches to gradient augmentation improve the *h*-ellipticity of the operator; for dual faces with zero skew angle, the edge-directional derivative, $\partial^e U$, is the only contributor. Hasselbacher [7] considered both formulations but used the EN formulation in computations. The FT formulation is identical to the approach used in a sheared mapped quadrilateral grid, i.e., the gradient is recovered from directional gradients in the mapped coordinate directions.

The FT formulation has been found to be more robust for highly-skewed grids and was used for cell-centered applications in [4]. The rationale is that, in such applications, the relative contributions from the edge gradient to the diffusion operator are much larger than with the EN formulation. Comparing (12) and (13), the flux contribution of $\partial^e U$ with EN augmentation is $\cos\theta$ (less than 1) versus $1/\cos\theta$ (greater than 1) with FT augmentation. Likewise, any contributions from $\partial^e U$ with the EN formulation vanish for θ approaching $\pi/2$. The face-normal gradient, entirely neglecting the projected Green–Gauss gradient, is shown in Fig. 3; the differences in the diffusion operator are easily seen to be a factor of two corresponding to the particular value of $\theta = \pi/4$.

The skew angle can approach $\pi/2$ on primal grids and even exceed $\pi/2$ on agglomerated grids, resulting in a destabilizing edge contribution for both approaches to augmentation. We have elected to neglect the entire flux at faces with $\theta \geqslant \pi/2$. An alternate approach, implemented as yet only in 2D, is to simply discard the directional derivative contribution.

2.2. Element-free discretizations

Two element-free discretizations are described below; at a minimum, they are needed in multigrid because the element-based data structures are not retained on agglomerated grids. Additionally, they can be used on the target grids—either to reduce computational cost or serve as drivers in relaxation.

Referring to Fig. 1, the element-free schemes approximate the integral flux through the dual faces adjacent to the edge [0,1] as

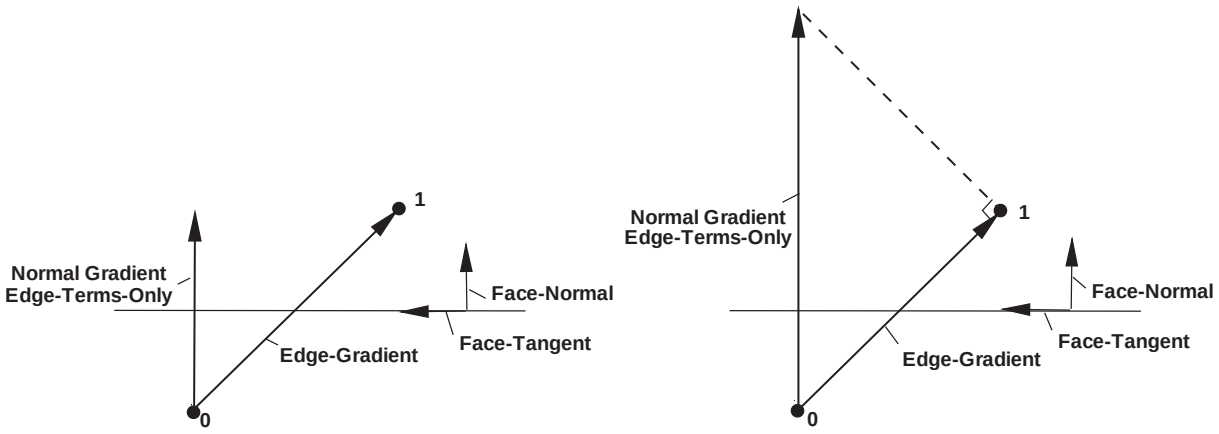
$$\int_{A\mu B} \nabla U \cdot \hat{\mathbf{n}}\, ds = \nabla U_\mu \cdot \mathbf{n}_\mu, \tag{14}$$

where the directed area, $\mathbf{n}_\mu$, is a lumped approximation,

$$\mathbf{n}_\mu = \mathbf{n}_{A\mu} + \mathbf{n}_{\mu B}. \tag{15}$$

The first scheme to approximate ∇U_μ, herein referred as *Edge-Terms-Only* (ETO), has already been introduced (Fig. 3) and is often referenced in the literature as a *thin-layer approximation*. Both edge-normal, ETO (EN), and face-tangent, ETO (FT), constructions can be used—either can be considered a thin-layer scheme. The gradient ∇U_μ is constructed using the right sides of either (10) or (11) retaining only the contributions from the $\partial^e U$ terms. The scheme is a positive scheme but on non-orthogonal grids (non-zero skew

Author's personal copy



(a) Edge-normal construction. (b) Face-tangent construction.

Fig. 3. Illustration of gradient constructions at a control-volume face separating nodes 0 and 1 using only edge gradients.

angles), it is not consistent (i.e., discrete solutions do not converge to the exact continuous solution with consistent grid refinement) [3,5,13,14]. The inconsistencies are most noticeable on grids with persistently-high skew angles—high-aspect-ratio tetrahedral meshes, for example.

The second scheme is the *average-least-squares* (Avg-LSQ) scheme. The gradient ∇U_μ is constructed using the right sides of either (10) or (11) with the gradient $\overline{\nabla U}$ replaced by the average of the least-squares (LSQ) gradients computed at the two nodes associated with the edge. The stencil of the LSQ gradient at a node includes all edge-connected neighbors. The LSQ minimization enforces the given solution at the central node.

3. Agglomerated grids

The control volumes of each agglomerated grid are found by summing control volumes of a finer grid. Any agglomerated grid can be defined in terms of a conservative agglomeration operator, R_0, as

$$\Omega^c = R_0 \Omega^f, \tag{16}$$

where superscripts c and f denote entities on coarser and finer grids, respectively. On the agglomerated grids, the control volumes become geometrically more complex than their primal counterparts and the details of the control-volume boundaries are not retained. The directed area of a coarse-grid face separating two agglomerated control volumes, if required, is found by lumping the directed areas of the corresponding finer-grid faces and is assigned to the virtual edge connecting the centers of the neighboring agglomerated control volumes.

As described more fully in [13], the grids are agglomerated within a topology-preserving framework, in which hierarchies are assigned based on connections to the computational boundaries and surface discontinuities. Corners are identified as grid points with three or more boundary-condition-type closures (or two or more boundary slope discontinuities). Ridges are identified as grid points with two boundary-condition-type closures (or one boundary slope discontinuity). Valleys are identified as grid points with a single boundary-condition-type closure and interiors are identified as grid points with no boundary closure. The agglomerations proceed hierarchically from seeds within the topologies, first corners, then ridges, then valleys, and finally interiors. Rules are enforced to maintain the boundary condition types of the finer grid within the agglomerated grid. For example, a ridge can be agglomerated into an existing ridge agglomeration only if the two boundary conditions associated with each ridge are the same. Hierarchies on each agglomerated grid are inherited from the finer grid.

There are two main difficulties associated with the current agglomeration techniques. The first is that after agglomeration, there may be insufficient connections to construct the least-square gradient at a node. This occurs most often near boundaries and, to improve reliability for complex geometries, we have adopted a boundary agglomeration step, in which corners, ridges, and valleys are agglomerated first—but agglomerations are allowed only within the same hierarchy. Thus, corners are never agglomerated. Ridges can be agglomerated only with ridges and valleys can be agglomerated only with valleys. These rules guarantee a valid non-degenerate LSQ stencil near boundaries. The downside is that the agglomerated grids have volumes near features much smaller than the interior volumes, especially on coarser grids. A better approach, implemented as yet only in 2D, is to augment the edge-connections as needed to construct gradients at a control volume.

The second difficulty, occurring more frequently in 3D than in 2D, is that large skew angles ($\theta \geqslant \pi/2$) are encountered on agglomerated grid faces. As discussed earlier, we neglect the entire flux at these faces in 3D. Another possible strategy is to control the shape of the agglomerations, either during agglomeration or in a post-processing step, in order to avoid extreme face skewness.

Typical isotropic grids are shown in Figs. 4 and 5, corresponding to a target grid and a first-level agglomeration, respectively. The target grids are all tetrahedral grids and are irregular because of 3D random node perturbations. The grids were constructed in a cubic domain and then mapped onto an ellipsoid. In the cubic domain, the grids are perturbed in each coordinate direction with magnitude 1/4 of the local mesh spacing.

Typical stretched grids are shown in Figs. 6 and 7. A prismatic layer is first generated from a triangulated boundary; the boundary grids include random node perturbations within the boundary surface. The prismatic layer occupies the lower quarter of the domain for all grid sizes. The maximum aspect ratio of 10^3 is enforced for cells at the bottom, where the aspect ratio is defined as a ratio of the mesh spacings tangent and normal to the boundary. Nodes in the prismatic layer were generated by a geometric sequence such that the aspect ratio approaches unity at line terminations. The

Author's personal copy

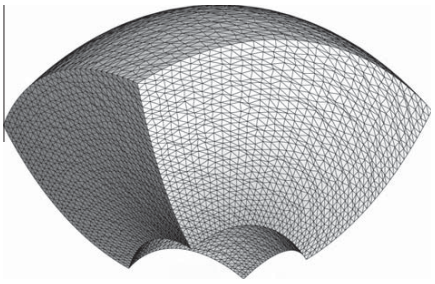


Fig. 4. Target isotropic $33 \times 33 \times 33$ grid.

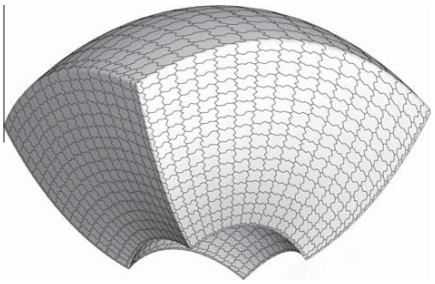


Fig. 5. First-level agglomeration generated from the target isotropic $33 \times 33 \times 33$ grid.

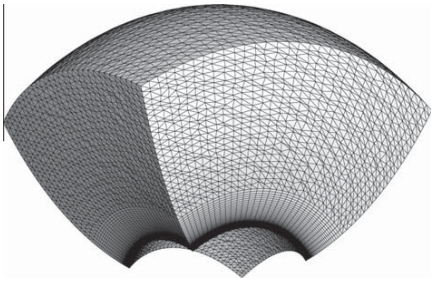


Fig. 6. Target stretched $33 \times 33 \times 134$ grid.

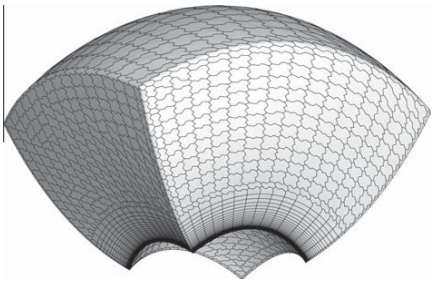


Fig. 7. First-level agglomeration generated from the target isotropic $33 \times 33 \times 134$ grid.

number of nodes per line is thus automatically determined. An isotropic tetrahedra grid with random 3D node perturbations is then added.

For highly stretched meshes, the advancing front agglomeration is first applied at the boundary of the grid (corners, ridges, and valleys) containing the origins of the implicit lines. Then interior duals are agglomerated, two at a time in the normal direction, from the boundary to the line terminations, preserving the prismatic structure of the agglomerations. After the line agglomerations, the front agglomeration method is applied over the remainder of the domain. The overall agglomeration technique is similar to that of Hyams et al. [8]

For both isotropic and stretched grids, a sequence of 15 target grids were generated to assess multigrid convergence. In Appendix A, details of the sequences are given and additional statistics for two grids are given.

4. Multigrid

Elements of the multigrid algorithm are presented in this section. A V-cycle [18], denoted as $V(\nu_1, \nu_2)$, uses ν_1 relaxations performed at each grid before proceeding to the coarser grid and ν_2 relaxations after coarse-grid correction; the coarsest grid is solved exactly (with many relaxations). Residuals, r^f, corresponding to the fine-grid discretization of the integral Eq. (2) are restricted to the coarse grid using the conservative agglomeration operator R_0, defined in (16), and a residual-averaging operator, W, as

$$r^c = R_0 W r^f. \tag{17}$$

The residual averaging is performed by replacing the individual residual at a node by the arithmetic average of the residuals over its neighbor nodes. For simplicity of implementation, the averaging is not performed over boundary nodes or nodes that connect to a boundary. Note that averaging, e.g., full-weighting, of residuals is necessary with multicolor relaxation schemes even in classical structured-grid multigrid methods because the residuals of the last color are reduced identically to zero. The fine-grid solution approximation is restricted to the coarse grid as

$$U^c = \frac{R_0(U^f \Omega^f)}{\Omega^c}. \tag{18}$$

The prolongations P_0 and P_1 are exact for piecewise-constant and linear functions, respectively. The prolongation P_0 is the transpose of R_0. The operator P_1 is constructed locally using linear interpolation from a triangle (2D) or tetrahedra (3D) defined on the coarse grid. The geometrical shape is anchored at the coarser-grid location of the agglomerate that contains the given finer control volume. Other nearby points are found using the adjacency graph. An enclosing simplex is sought that avoids prolongation with non-convex weights and, in situations where multiple geometrical shapes are found, the first one encountered is used. At locations where this procedure results in non-convex weights, the prolongation is reverted locally to piecewise-constant prolongation. The prolongation operator P_1 is modified to prolong only from hierarchies equal or above the hierarchy of the prolonged point. The correction δU to the finer grid is prolonged typically through P_1, as

$$(\delta U)^f = P_1(\delta U)^c. \tag{19}$$

The available target-grid and coarse-grid discretizations are listed in Table 1. The main target discretization of interest is the element-based Green–Gauss scheme discussed earlier with either of the two approaches to gradient augmentation for non-simplicial elements. There are four available element-free coarse-grid discretizations, the consistent Avg-LSQ scheme and the inconsistent but

Table 1
Summary of target-grid and coarse-grid discretizations; gradient augmentation is denoted in parentheses.

Target-grid discretization	Coarse-grid discretization
Green–Gauss (EN)	Avg-LSQ (EN)
	ETO (EN)
Green–Gauss (FT)	Avg-LSQ (FT)
	ETO (FT)

Author's personal copy

widely-used ETO scheme, each evaluated with the same approach to gradient augmentation used on the target grid for simplicity.

The exact linear operator is used in the iterative phase of the Green–Gauss scheme, enabling a robust multicolor Gauss–Seidel relaxation. The Avg-LSQ scheme has a comparatively larger stencil and its exact linearization is not used in iterations; instead relaxation of the Avg-LSQ scheme relies on the ETO linearization as a driver. It is known that the smoothing rate with this approach can deteriorate on highly-skewed grids [4].

5. Analysis

5.1. Idealized relaxation and idealized coarse grid methods

This section presents quantitative analysis tools, idealized relaxation (IR) and idealized coarse-grid (ICG) iterations, for assessment and improvement of unstructured multigrid solvers. IR and ICG have been applied earlier [13] to analyze multigrid solvers on isotopic unstructured grids; applications to high-aspect-ratio grids are studied below.

It is long known [18] that convergence of full-coarsening multigrid with point relaxation deteriorates on grids with high aspect ratio. Failure of point relaxation to smooth errors oscillating in the direction of weak coupling (larger mesh spacing) is the main reason for convergence deterioration. Typical remedies involve implicit relaxation, semi-coarsening, or a combination of the two. In this paper, multigrid employs full-coarsening and line-implicit relaxation.

Testing of multigrid solvers with line-implicit relaxation schemes on high-aspect-ratio grids is not straightforward. At the initial design stages, the performance of a multigrid cycle is typically tested on either small low-density grids or with Dirichlet conditions imposed at boundaries of the high-aspect-ratio regions. On such grids, a line-implicit relaxation scheme becomes a solver rather than a smoother and provides overly optimistic predictions [18]. IR and ICG cycles, similarly to LFA, avoid this difficulty and can expose problems that may arise only in applications with extremely large numbers of degrees-of-freedom.

Specifically the IR and ICG methods focus on the main complementary parts of a multigrid cycle: relaxation and coarse-grid correction. Each part of the cycle is assigned a task, e.g., relaxation is typically assigned to smooth errors, coarse-grid correction is typically assigned to reduce all smooth error components. In the analysis, idealized iterations probe the actual two-grid cycle to identify parts limiting the overall effectiveness.

The IR and ICG iterations can be applied to any formulation with a manufactured solution; here they are applied to a formulation with zero manufactured solution. The initial guess is formed by a random perturbation of the solution. In the analysis, one part of the tested cycle is replaced with an idealized imitation. The idealized imitations do not depend on the operators to be solved. Rather, they are numerical procedures acting directly on the known algebraic error to fulfill the task assigned to the corresponding part of the two-grid cycle. The results of the analysis are convergence patterns of the iterations that may either confirm or refute expectations as to how well each part of the actual cycle is carrying out the assigned task.

With IR cycles, the coarse-grid correction part is actual and the relaxation is idealized. Idealized relaxation can be implemented by constructing a pseudo-Laplacian operator, A^{IR}, which includes nodes linked by an edge, or possibly an element through a virtual edge, to a given node, as below,

$$A^{IR}\epsilon \equiv \sum_{i=1}^{N_e} w_i(\epsilon_i - \epsilon_0) = 0. \tag{20}$$

Here, N_e is the number of edges connected to node 0, the algebraic error at node i is ϵ_i, and w_i represents a weight. The choice $w_i = 1$ yields a positive operator. A few relaxations of (20) serve as an idealized relaxation.

With ICG cycles, the relaxation scheme is actual and the coarse-grid correction is idealized. The ICG correction used for unstructured multigrid computations is defined in the following two steps: (1) The algebraic error is restricted to the coarse grid by a volume-averaging operator, similarly to the solution restriction (18). (2) The volume-averaged error is interpolated back to the fine grid as a correction. This procedure effectively reduces all smooth error components.

An important check of the quality of chosen idealized components is convergence of the "reference cycle," which uses both idealized components in iterations. The convergence rate of the reference cycle represents a sensitivity threshold in that idealized iterations generally suggest some meaningful improvements only for actual cycles with convergence rates significantly slower than this threshold.

The idealizations used in IR and ICG analysis are not unique. Within high-aspect-ratio grid regions, we consider a line-implicit IR scheme, designated IR-L, that simultaneously changes algebraic errors at all nodes of the same grid line such that the updated algebraic errors satisfy (20); the lines are visited in a 2-color order. The

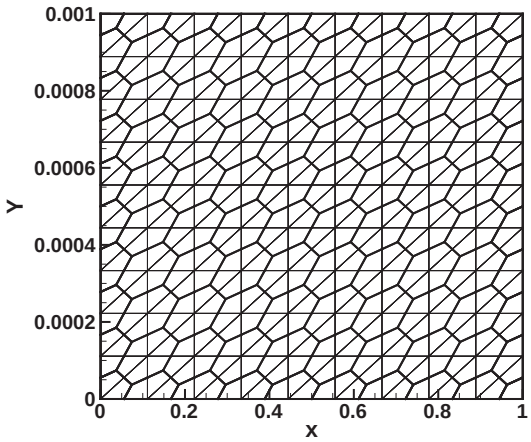


Fig. 8. Control volume boundaries (heavier lines) for regular triangular fine grid.

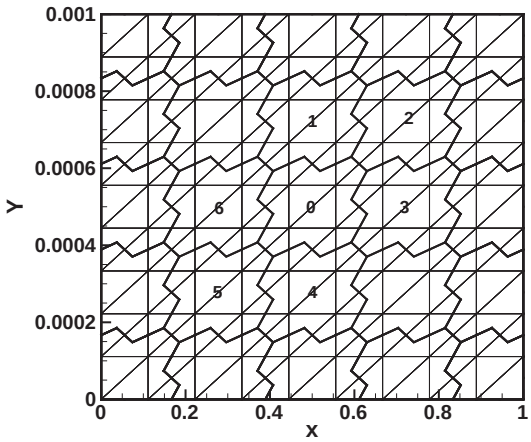


Fig. 9. Control volume boundaries (heavier lines) for regular triangular coarse grid.

Author's personal copy

selection is justified through LFA of regular quadrilateral and triangular grids in Appendices B and C. Details of the LFA methodology are summarized in Appendix B. Several point- and line-implicit idealized relaxations performed in various orders are analyzed in Appendix C. Within isotropic grid regions, an idealized relaxation with multicolor point-wise error averaging, designated IR-P, is used. Appendix D presents observations on convergence rates of IR-P and actual cycles on isotropic unstructured grids. The two idealized relaxations, IR-P and IR-L, overlap by a single node per line for stretched grids including isotropic and high-aspect-ratio regions.

5.2. Applications to triangular grids

Illustrative 2-grid computations are performed on a sequence of regular triangular grids with uniform aspect ratio $A = 10^3$. Fine-grid and coarse-grid control volumes are illustrated in Figs. 8 and 9. Note that on the fine grid, the Green–Gauss discretization is equivalent to a classical 5-point Laplacian [4].

Table 2 shows asymptotic convergence rates with IR-L and residual averaging for various coarse-grid discretizations. We do not show actual relaxations because Dirichlet conditions were used in the computations and the line-implicit relaxation solves the equations in a single iteration. For comparisons with the rates one would observe in computations on large grids, Table 3 shows convergence rates computed with LFA using methodology presented in Appendix B. On regular grids, LFA is known to provide accurate predictions of multigrid convergence.

All analysis methods indicate that only discretizations with EN augmentation allow fast grid-independent convergence on high-aspect-ratio triangular grids. Convergence of multigrid with coarse-grid discretizations using FT augmentation approaches unity in the limit of grid refinement.

The reason for the striking differences between EN and FT approaches to augmentation can be traced directly to the high skewing of the coarse grid shown in Fig. 9. Considering a fully-interior control volume, there are six face-connections to the surrounding control volumes. Two of these faces (connecting node 0 with nodes 1 and 4, respectively, in Fig. 9) have nearly-zero skew angle and the other four faces have skew angles approaching $\pi/2$. Considering the discrete diffusion terms in the y-direction, the coarse-grid ETO (EN) operator is inconsistent, being 5/6 of the fine-grid operator. However, this is sufficient to yield a convergence rate of 0.2 per multigrid cycle. The coarse-grid Avg-LSQ (EN) scheme is consistent and provides an order of magnitude error reduction per multigrid cycle. Additional details and specific formulas are provided in Appendix B.

Schemes with the FT gradient augmentation magnify the skewed-face contributions to the diffusion operator. The Avg-LSQ (FT) scheme leads to a wide-stencil coarse-grid operator, poorly approximating the fine-grid medium-range error components oscillating in the x-direction. The ETO (FT) scheme leads to completely inaccurate approximations (see additional details in Appendix B).

In these regular-grid computations, the control-volume centers on the coarse grid remain perfectly collinear. In general, any departure from the perfect alignment, such as with an irregular triangularization of the fine grid and a volume-weighted construction of the coarse-grid control-volume locations, can result in high skew angles at all faces. In this situation, the ETO (EN) scheme becomes inadequate. The Avg-LSQ (EN) scheme loses h-ellipticity and, at a minimum, becomes difficult to converge.

All of the above issues associated with highly-skewed faces on high-aspect-ratio grids are avoided if prismatic (quadrilateral in 2D) grids are used with the line agglomeration discussed earlier. The skewness of the coarse grid is then comparable with the skewness of the fine grid and convergence rates for all schemes become an order of magnitude per cycle. Thus, only prismatic grids are used in highly-stretched regions for the computations that follow.

Table 2
Asymptotic convergence rates for IR-L cycle; regular triangular grid; $\nu_1 = \nu_2 = 2$.

Fine grid	Avg-LSQ (EN)	ETO (EN)	Avg-LSQ (FT)	ETO (FT)
32×32	<0.1	0.16	0.13	0.32
64×64	<0.1	0.16	0.28	0.56
128×128	<0.1	0.18	0.44	0.73

Table 3
LFA 2-grid convergence rates for IR-L and actual line-implicit cycles; regular triangular grid; $\nu_1 = \nu_2 = 2$; piecewise-constant prolongation.

LFA	Avg-LSQ (EN)	ETO (EN)	Avg-LSQ (FT)	ETO (FT)
IR-L	0.12	0.20	1.0	1.0
Actual	0.07	0.19	1.0	1.0

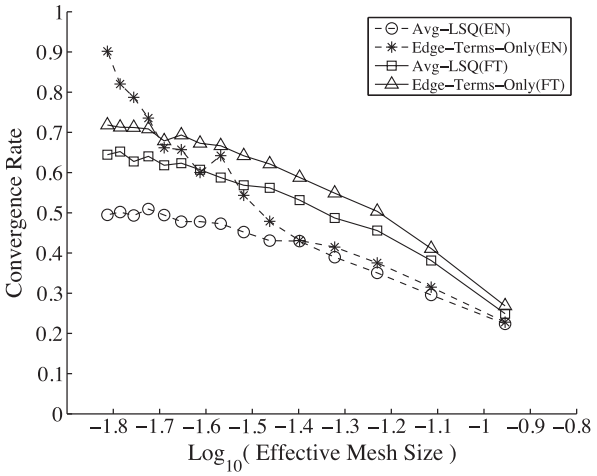


Fig. 10. Convergence rate versus effective mesh size for isotropic grids; $\nu_1 = 2; \nu_2 = 1$.

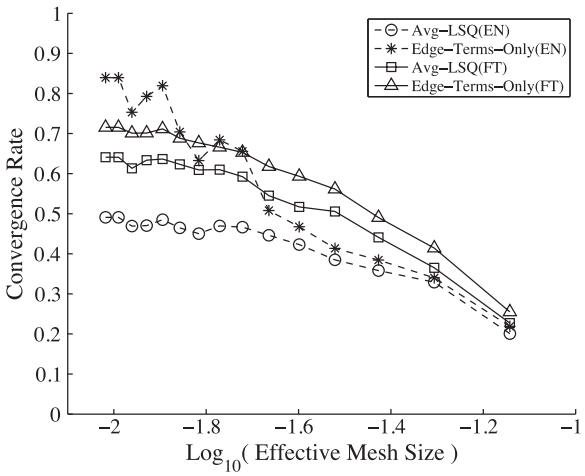


Fig. 11. Convergence rate versus effective mesh size for stretched grids; $\nu_1 = 2; \nu_2 = 1$.

Author's personal copy

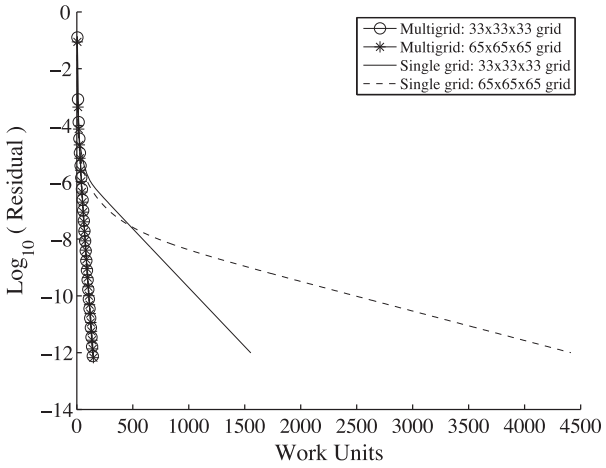


Fig. 12. Convergence versus work units for two isotropic grids; $v_1 = 2; v_2 = 1$; coarse-grid discretization is the Avg-LSQ (EN) scheme.

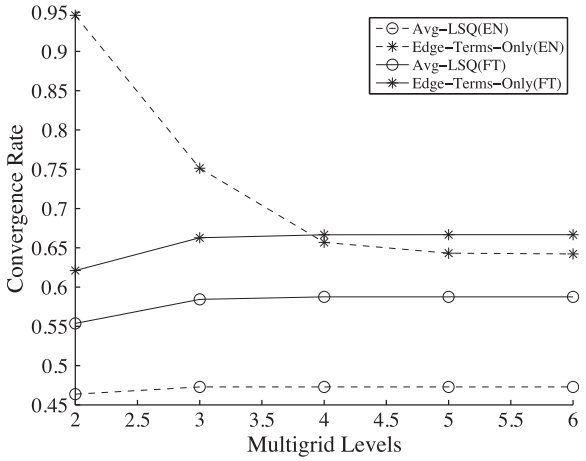


Fig. 14. Convergence rates versus multigrid levels for a $37 \times 37 \times 37$ isotropic grid; $v_1 = 2; v_2 = 1$.

6. Three-dimensional results

In this section, we present 3D multigrid convergence rates for the sequences of isotropic and stretched grids listed in Appendix A; for each of the 15 grids in the sequence, multigrid employs all available levels. Initial conditions on each grid were taken as random and the convergence was terminated when integral-equation residuals reached machine-precision level. Figs. 10 and 11 show multigrid convergence rates versus the effective mesh size for each of the coarse-grid discretizations. The effective mesh size is defined as the reciprocal of the cube root of the total number of nodes. The convergence rate is computed as an average of per-cycle convergence rates over the last four multigrid cycles. In grid refinement, the convergence rates approach grid-independent levels for the Avg-LSQ (EN), Avg-LSQ (FT), and ETO (FT) schemes; the best convergence rate is obtained with the Avg-LSQ (EN) scheme. Observe that the convergence with these schemes for stretched grids is as good as convergence for isotropic grids.

To demonstrate the essentially grid-independent convergence with the Avg-LSQ (EN) coarse-grid discretization, single-grid and multigrid computations are compared in Figs. 12 and 13 for isotropic and stretched grids, respectively. Convergence for two grids, one finer by a factor of two in each direction, are shown. The integral-equation residual is shown versus work units, taken as the number of residual evaluations on the fine grid. For the current Full Approximation Scheme [18] multigrid implementation, the work units per cycle are estimated as $(v_1 + v_2 + 2)(1 + 1/8 + 1/64 + \cdots)$. The results show the expected slowdown of the single-grid scheme on the finer grid. The finer-grid residual convergence over-plots that of the coarser grid with the multigrid scheme.

Multigrid convergence of the ETO (EN) scheme is highly grid-dependent, slowing down on finer grids for both isotropic and stretched grids. These results confirm the conclusions drawn from the previous study [13] for isotropic tetrahedra on cubical domains—multigrid convergence is grid-dependent with the ETO (EN) scheme and grid-independent with the Avg-LSQ (EN) scheme.

During the numerical experiments, it was observed that, contrary to usual expectations, multigrid with the ETO (EN) scheme converges better with multiple levels than with two levels (the coarsest problem is fully solved in all cases). Figs. 14 and 15 show convergence rates versus multigrid levels for the two grids listed in Tables 5 and 6. The existence of faces with skew angles greater that $\pi/2$ do not appear to have a negative impact on convergence for the Avg-LSQ schemes; 2-level convergence is comparable with multilevel convergence. It is not surprising that multigrid with the

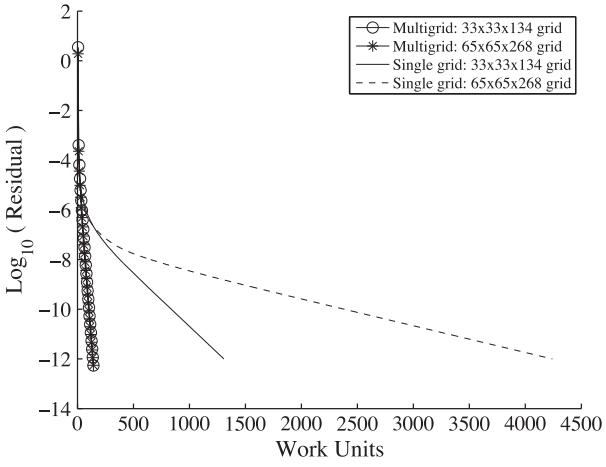


Fig. 13. Convergence versus work units for two stretched grids; $v_1 = 2; v_2 = 1$; coarse-grid discretization is the Avg-LSQ (EN) scheme.

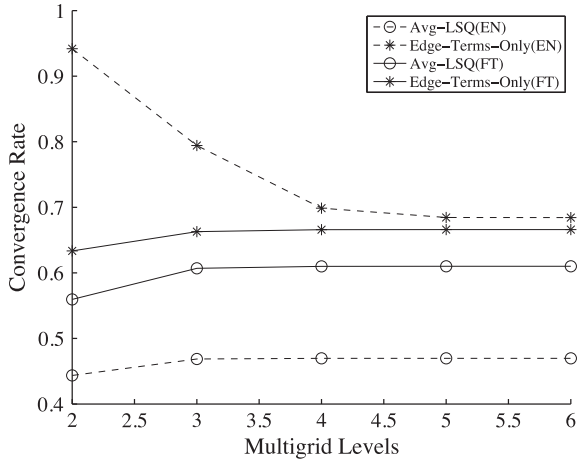


Fig. 15. Convergence rates versus multigrid levels for a $37 \times 37 \times 150$ stretched grid; $v_1 = 2; v_2 = 1$

Author's personal copy

ETO (EN) schemes exhibits grid-dependent convergence because the scheme is inconsistent. What is surprising is that the ETO (FT) scheme does not fail (see Figs. 10 and 11). Although we do not show the results here, for more realistic complex geometries, we have found that multigrid with either ETO scheme fails to converge.

7. Conclusions

Agglomerated multigrid techniques used in unstructured-grid methods have been critically studied for a model problem representative of laminar diffusion in the incompressible limit, with a focus on highly-stretched grids. A multigrid solver for a node-centered element-based discretization has been investigated with several different coarse-grid discretizations on agglomerated grids. Quantitative analysis methods have been used to identify and assess elements of the solver that perform well in high-aspect-ratio regions. The elements of multigrid enabling grid-independent convergence rates are the following: (1) a consistent coarse-grid discretization; (2) prismatic elements with line relaxation and line agglomeration in the stretched grid regions; and (3) residual averaging of the conservative residuals before restriction. The convergence rates per cycle on mixed-element grids with highly-stretched regions are commensurate with the convergence rates on isotropic grids.

Analyses and computations show that multigrid convergence severely degrades with inconsistent ETO coarse-grid discretizations. On regular simplicial high-aspect-ratio grids, analyses show that the Avg-LSQ (FT) coarse-grid discretization leads to convergence deterioration. On irregular simplicial high-aspect-ratio grids, convergence of multigrid with the Avg-LSQ (EN) coarse-grid discretization is also expected to deteriorate. Using other coarse-grid discretizations with simplicial elements in highly-stretched regions may be possible but is not straightforward and requires further study.

Acknowledgments

The results were computed with the FUN3D suite of codes (http://fun3d.larc.nasa.gov/) at NASA Langley Research Center.

Appendix A. Agglomerated grid details

Table 4 lists grid sizes and numbers of grids agglomerated for the target grid sequences generated to assess multigrid convergence. For the stretched grids, the number of nodes in each implicit line is also listed. Tables 5 and 6 show the maximum skew angle and the coarsening ratio of each agglomeration level for two typical grids. The coarsening ratio is defined as the number of finer-grid degrees-of-freedom divided by the number of degrees-of-freedom at a given coarse level, ideally approaching 8 for full-coarsening in 3D. The coarsening ratio is above 6 on the first agglomeration but degrades on coarser levels. Note, for reference, that the isotropic tetrahedral meshes have a maximum skew angle of approximately 75° and that faces with skew angles greater that $\pi/2$ are encountered on the fourth level for the isotropic grid and on the fifth level for the stretched grid.

Table 4
Grid sizes for isotropic and stretched grids; the first number in parenthesis is the numbers of agglomerated grids; the second number in parentheses is the number of nodes per implicit line.

Isotropic grids	Stretched grids
09 × 09 × 09 (2)	09 × 09 × 33 (2.26)
13 × 13 × 13 (3)	13 × 13 × 49 (3.39)
17 × 17 × 17 (4)	17 × 17 × 66 (4.53)
21 × 21 × 21 (4)	21 × 21 × 83 (4.67)
25 × 25 × 25 (5)	25 × 25 × 100 (5.81)
29 × 29 × 29 (5)	29 × 29 × 117 (5.95)
33 × 33 × 33 (6)	33 × 33 × 134 (6.109)
37 × 37 × 37 (6)	37 × 37 × 150 (6.122)
41 × 41 × 41 (6)	41 × 41 × 167 (6.136)
45 × 45 × 45 (6)	45 × 45 × 184 (6.150)
49 × 49 × 49 (7)	49 × 49 × 201 (7.164)
53 × 53 × 53 (7)	53 × 53 × 218 (7.178)
57 × 57 × 57 (7)	57 × 57 × 235 (7.192)
61 × 61 × 61 (7)	61 × 61 × 251 (7.205)
65 × 65 × 65 (7)	65 × 65 × 268 (7.219)

Table 5
Maximum skew angle (°) and coarsening ratio of each agglomeration level for the 37 × 37 × 37 isotropic grid.

Level	Maximum skew angle (°)	Coarsening ratio
2	79.8	6.3
3	81.9	5.5
4	96.8	4.2
5	88.3	3.0
6	89.8	2.2

Table 6
Maximum skew angle (°) and coarsening ratio of each agglomeration level for the 37 × 37 × 150 stretched grid.

Level	Maximum skew angle (°)	Coarsening ratio
2	72.2	6.6
3	78.1	5.8
4	78.7	4.6
5	91.9	3.6
6	89.2	2.8

Appendix B. Local Fourier analysis for regular grids

Asymptotic convergence rates of 2-grid cycles are predicted using LFA on regular 2D triangular and quadrilateral grids. Details pertaining to the analysis are given below. Foundations and applications of LFA can be found in the original paper [6] and in textbooks, e.g. [18]. The Green–Gauss discretization scheme is used on the fine grids; as noted earlier, for these fine grids, the scheme is the five-point Laplacian operator. Interior control-volume boundaries on a regular triangular fine grid are illustrated in Fig. 8. The coarse-grids schemes are applied on fully-coarsened agglomerated coarse grids (interior coarse-grid control volumes corresponding to Fig. 8 are illustrated in Fig. 9).

The Fourier symbol of a 2-grid cycle, $\widehat{M}$, is a 4×4 matrix acting in the linear vector space corresponding to the amplitudes of the following quartet of Fourier components,

$$e^{i(\theta_x i_x + \theta_y i_y)}, \quad e^{i((\theta_x+\pi) i_x + \theta_y i_y)}, \quad e^{i(\theta_x i_x + (\theta_y+\pi) i_y)}, \quad e^{i\left((\theta_x+\pi) i_x + (\theta_y+\pi) i_y\right)}, \tag{B.1}$$

with horizontal and vertical node indexes, i_x and i_y, respectively, and normalized Fourier frequencies

$$\begin{aligned} \bar{\theta}^1 &= \left(\theta_x^1, \theta_y^1\right) = (\theta_x, \theta_y), \\ \bar{\theta}^2 &= \left(\theta_x^2, \theta_y^2\right) = (\theta_x + \pi, \theta_y), \\ \bar{\theta}^3 &= \left(\theta_x^3, \theta_y^3\right) = (\theta_x, \theta_y + \pi), \\ \bar{\theta}^4 &= \left(\theta_x^4, \theta_y^4\right) = (\theta_x + \pi, \theta_y + \pi) \end{aligned} \tag{B.2}$$

satisfying $\max(|\theta_x|, |\theta_y|) \leqslant \pi/2$.

Author's personal copy

Table 7
Main-diagonal symbols of lexicographic relaxations; actual is line-implicit relaxation.

Relaxation	Fine grid	$\widehat{S}^k \equiv N/D$
IR-P	Quad.	$N = e^{i\theta_x^k} + e^{i\theta_y^k}$ $D = 4 - e^{-i\theta_x^k} - e^{-i\theta_y^k}$
IR-P	Tria.	$N = e^{i\theta_x^k} + e^{i\theta_y^k} + e^{i(\theta_x^k+\theta_y^k)}$ $D = 6 - e^{-i\theta_x^k} - e^{-i\theta_y^k} - e^{-i(\theta_x^k+\theta_y^k)}$
IR-L	Quad.	$N = e^{i\theta_x^k}$ $D = 4 - e^{-i\theta_x^k} - 2\cos(\theta_y^k)$
IR-L	Tria.	$N = e^{i\theta_x^k} + e^{i(\theta_x^k+\theta_y^k)}$ $D = 6 - e^{-i\theta_x^k} - 2\cos\left(\theta_y^k\right) - e^{-i(\theta_x^k+\theta_y^k)}$
Actual	Either	$N = e^{i\theta_x^k}$ $D = 2 + 2A^2 - e^{-i\theta_x^k} - 2A^2\cos(\theta_y^k)$

$$\widehat{M} = \widehat{S}^{\nu_2}\left(\widehat{E} - \widehat{P}\widehat{L}_H^{-1}\widehat{R}\widehat{W}\widehat{L}_h\right)\widehat{S}^{\nu_1}. \tag{B.3}$$

Here, $\nu_1 = \nu_2 = 2$ are the numbers of pre- and post-relaxation sweeps, $\widehat{S}$ is the relaxation operator symbol, $\widehat{L}_h$ and $\widehat{L}_H$ are the fine- and coarse-grid operator symbols, $\widehat{W}$ is the residual-averaging operator symbol, $\widehat{P}$ and $\widehat{R}$ are the symbols of the prolongation and restriction operators corresponding to P_0 and R_0, respectively, and $\widehat{E}$ is the 4×4 identity matrix.

The symbols $\widehat{L}_h$ and $\widehat{W}$ are 4×4 diagonal matrices and the symbols $\widehat{R}$ and $\widehat{P}$ are 1×4 and 4×1 vectors, respectively, each composed of scalar Fourier symbols. The scalar symbols are computed for each of the components (B.1). The diagonal entries of the fine-grid operator symbol, $\widehat{L}_h$, are

$$\widehat{L}_h^k = \frac{2}{h_y^2}\left[\left(-1 + \cos\left(\theta_y^k\right)\right) + \frac{1}{A^2}\left(-1 + \cos\left(\theta_x^k\right)\right)\right], \tag{B.4}$$

where h_y and $h_x = A\,h_y$ are fine-grid mesh spacings in the corresponding directions and A is the grid aspect ratio. The symbols $\widehat{R}$ and $\widehat{P}$ relate the amplitudes of the four fine-grid Fourier components (B.1) to the amplitude of the corresponding coarse-grid Fourier component $e^{i2(\theta_x i_x + \theta_y i_y)}$ and assume that the coarse grid node (i_x, i_y) is located at the center of the rectangle formed by the four fine-grid nodes $(2i_x, 2i_y)$, $(2i_x + 1, 2i_y)$, $(2i_x, 2i_y + 1)$, and $(2i_x + 1, 2i_y + 1)$. The entries of $\widehat{R}$ are

$$\widehat{R}^k = \frac{1}{4}\left(1 + e^{i\theta_x^k} + e^{i\theta_y^k} + e^{i\left(\theta_x^k+\theta_y^k\right)}\right). \tag{B.5}$$

The entries of $\widehat{P}$ are

$$\widehat{P}^k = \frac{1}{4}\left(1 + e^{-i\theta_x^k} + e^{-i\theta_y^k} + e^{-i\left(\theta_x^k+\theta_y^k\right)}\right). \tag{B.6}$$

The entries of $\widehat{W}$ are shown below for triangular and quadrilateral grids,

$$(\widehat{W}^k)_{\text{tria.}} = \frac{1}{3}\left(\cos(\theta_x^k) + \cos(\theta_y^k) + \cos(\theta_x^k + \theta_y^k)\right), \tag{B.7}$$

$$(\widehat{W}^k)_{\text{quad.}} = \frac{1}{2}\left(\cos(\theta_x^k) + \cos(\theta_y^k)\right). \tag{B.8}$$

The symbols of relaxations performed in the lexicographic order are 4×4 diagonal matrices composed of scalar Fourier symbols. Table 7 shows the main-diagonal symbols for lexicographic-order idealized and actual relaxations.

Multicolor relaxations depend on the specific relaxation order and their symbols are 4×4 matrices with a more complex structure. For example, the symbol of a 2-color line relaxation has a block-diagonal structure with two 2×2 diagonal blocks; the block corresponding to the frequencies $\bar{\theta}^1$ and $\bar{\theta}^2$ is defined as

Table 8
Symbols of implicit-line Jacobi relaxation.

Operator	Fine grid	$\widehat{D}^k \equiv N/D$
IR	Tria.	$N = \cos(\theta_x^k) + \cos\left(\theta_x^k + \theta_y^k\right)$ $D = 3 - \cos\left(\theta_y^k\right)$
IR	Quad.	$N = \cos\left(\theta_x^k\right)$ $D = 2 - \cos\left(\theta_y^k\right)$
Actual	Either	$N = \cos(\theta_x^k)$ $D = 1 + A^2 - A^2\cos(\theta_y^k)$

$$\frac{1}{2}\begin{bmatrix} \widehat{D}^1(1+\widehat{D}^1) & \widehat{D}^2(1-\widehat{D}^2) \\ \widehat{D}^1(1-\widehat{D}^1) & \widehat{D}^2(1+\widehat{D}^2) \end{bmatrix}, \tag{B.9}$$

where the scalar symbol $\widehat{D}^k$, corresponding to line-implicit Jacobi relaxation, is given in Table 8 for the operators and grids considered.

To describe a 4-color relaxation, let color 1 mark points with i_x even and i_y even, color 2 mark points with i_x odd and i_y even, color 3 mark points with i_x even and i_y odd, and color 4 mark points with i_x odd and i_y odd. First, the point-amplification symbols, C_j^i, for each color are computed where subscripts and superscripts denote color and frequency, respectively. Table 9 collects point-amplification symbols for two multicolor IR-P schemes performed in the (1234) order. The relaxation symbol is the following matrix:

$$\widehat{S}(\bar{\theta}) = \frac{1}{4}\begin{bmatrix} C_1^1 + C_2^1 + C_3^1 + C_4^1 & C_1^2 - C_2^2 + C_3^2 - C_4^2 & C_1^3 + C_2^3 - C_3^3 - C_4^3 & C_1^4 - C_2^4 - C_3^4 + C_4^4 \\ C_1^1 - C_2^1 + C_3^1 - C_4^1 & C_1^2 + C_2^2 + C_3^2 + C_4^2 & C_1^3 - C_2^3 - C_3^3 + C_4^3 & C_1^4 + C_2^4 - C_3^4 - C_4^4 \\ C_1^1 + C_2^1 - C_3^1 - C_4^1 & C_1^2 - C_2^2 - C_3^2 + C_4^2 & C_1^3 + C_2^3 + C_3^3 + C_4^3 & C_1^4 - C_2^4 + C_3^4 - C_4^4 \\ C_1^1 - C_2^1 - C_3^1 + C_4^1 & C_1^2 + C_2^2 - C_3^2 - C_4^2 & C_1^3 - C_2^3 + C_3^3 - C_4^3 & C_1^4 + C_2^4 + C_3^4 + C_4^4 \end{bmatrix}. \tag{B.10}$$

The coarse-grid operator symbol, $\widehat{L}_H$, is a scalar function of the coarse-grid frequency, $\left(\theta_x^H, \theta_y^H\right) \equiv (2\theta_x, 2\theta_y)$, specific to the given coarse-grid discretization. With a quadrilateral fine grid, both coarse and fine grids are orthogonal and $\widehat{L}_H$ is defined as,

$$\widehat{L}_H = \frac{1}{2h_y^2}\left[-1 + \cos\left(\theta_y^H\right) + \frac{1}{A^2}\left(-1 + \cos\left(\theta_x^H\right)\right)\right]. \tag{B.11}$$

The operator $\widehat{L}_H^*$ is composed of the leading-order terms in an expansion of $\widehat{L}_H$, assuming small (θ_x, θ_y). For (B.11),

$$\widehat{L}_H^* = \frac{-1}{h_y^2}\left(\theta_y^2 + \frac{1}{A^2}\theta_x^2\right) \tag{B.12}$$

coincides with the differential operator applied to the Fourier component $e^{i(\theta_x x/h_x + \theta_y y/h_y)}$, thus demonstrating that the coarse-grid operator for quadrilateral fine grids is consistent. Table 10 collects the

Table 9
Point-amplification symbols in 4-color (1234) IR-P schemes.

Fine grid	Symbol
Quad.	$C_1^k = \frac{1}{2}\left(\cos\left(\theta_x^k\right) + \cos\left(\theta_y^k\right)\right)$ $C_2^k = \frac{1}{2}\left(C_1^k\cos\left(\theta_x^k\right) + \cos\left(\theta_y^k\right)\right)$ $C_3^k = \frac{1}{2}\left(\cos\left(\theta_x^k\right) + C_1^k\cos\left(\theta_y^k\right)\right)$ $C_4^k = \frac{1}{2}\left(C_3^k\cos\left(\theta_x^k\right) + C_2^k\cos(\theta_y^k)\right)$
Tria.	$C_1^k = \frac{1}{3}\left(\cos\left(\theta_x^k\right) + \cos\left(\theta_y^k\right) + \cos\left(\theta_x^k + \theta_y^k\right)\right)$ $C_2^k = \frac{1}{3}\left(C_1^k\cos\left(\theta_x^k\right) + \cos\left(\theta_y^k\right) + \cos\left(\theta_x^k + \theta_y^k\right)\right)$ $C_3^k = \frac{1}{3}\left(\cos\left(\theta_x^k\right) + C_1^k\cos\left(\theta_y^k\right) + C_2^k\cos\left(\theta_x^k + \theta_y^k\right)\right)$ $C_4^k = \frac{1}{3}\left(C_3^k\cos\left(\theta_x^k\right) + C_2^k\cos\left(\theta_y^k\right) + C_1^k\cos\left(\theta_x^k + \theta_y^k\right)\right)$

Author's personal copy

Table 10
Symbol of coarse-grid operators for triangular fine grids.

Scheme	$\widehat{L}_H\left(2h_y^2\right)$
ETO (EN)	$\frac{5}{6}\left(-1+\cos\left(\theta_y^H\right)\right)+\frac{5}{6}\frac{1}{A^2}\left(-1+\cos(\theta_x^H)\right)$ $+\frac{1}{3}\frac{1}{A^2+1}\left(-1+\cos\left(\theta_x^H+\theta_y^H\right)\right)$
Avg-LSQ (EN)	$[\widehat{L}_H]^{\mathrm{ETO(EN)}}\left(2h_y^2\right)$ $+\frac{1}{36A^2}\left[\left(-\sin(\theta_y^H)+2\sin\left(\theta_x^H\right)+\sin\left(\theta_x^H+\theta_y^H\right)\right)\right.$ $\left.\left(\sin(\theta_y^H)+\frac{A^2-1}{A^2+1}\sin\left(\theta_x^H+\theta_y^H\right)\right)\right]$ $+\frac{1}{36}\left[\left(2\sin\left(\theta_y^H\right)-\sin\left(\theta_x^H\right)+\sin\left(\theta_x^H+\theta_y^H\right)\right)\right.$ $\left.\left(\sin\left(\theta_x^H\right)-\frac{A^2-1}{A^2+1}\sin\left(\theta_x^H+\theta_y^H\right)\right)\right]$
ETO (FT)	$\left(\frac{5}{6}+\frac{1}{30}\frac{1}{A^2}\right)\left(-1+\cos\left(\theta_y^H\right)\right)$ $+\left(\frac{5}{6}\frac{1}{A^2}+\frac{1}{30}\right)\left(-1+\cos\left(\theta_x^H\right)\right)$ $+\frac{1}{12}\left(1+\frac{1}{A^2}\right)\left(-1+\cos\left(\theta_x^H+\theta_y^H\right)\right)$
Avg-LSQ (FT)	$[\widehat{L}_H]^{\mathrm{ETO(FT)}}\left(2h_y^2\right)$ $+\frac{1}{36A^2}\left[\left(-\sin\left(\theta_y^H\right)+2\sin\left(\theta_x^H\right)+\sin\left(\theta_x^H+\theta_y^H\right)\right)\right.$ $\left.\left(\sin\left(\theta_y^H\right)+\frac{A^2}{5}\sin\left(\theta_x^H\right)+\frac{A^2-1}{2}\sin\left(\theta_x^H+\theta_y^H\right)\right)\right]$ $+\frac{1}{36}\left[\left(2\sin\left(\theta_y^H\right)-\sin\left(\theta_x^H\right)+\sin\left(\theta_x^H+\theta_y^H\right)\right)\right.$ $\left.\left(\sin\left(\theta_x^H\right)+\frac{A^2}{5}\sin\left(\theta_y^H\right)-\frac{A^2-1}{2A^2}\sin\left(\theta_x^H+\theta_y^H\right)\right)\right]$

symbols of the coarse-grid operator corresponding to various coarse-grid discretizations on triangular fine grids. The symbols for the Avg-LSQ and ETO discretizations are shown for both EN and FT augmentations. Table 11 collects the corresponding expansions for $A = 1$ and in the limit of $A \to \infty$. With quadrilateral grids or with Avg-LSQ discretizations, the coarse-grid operators are consistent for all A. On triangular grids, both the ETO (EN) and ETO (FT) discretizations are inconsistent for all A.

The asymptotic convergence rates are computed as the maximum spectral radius of $\widetilde{M}$ over all possible Fourier frequencies. Since the maximum amplification on high-aspect-ratio grids is expected for frequencies extremely smooth in the y-direction ($|\theta_y| \approx 0$), the frequency domain $(\theta_x, \theta_y) \in [-\pi, \pi]^2$ is, first, searched with the increment 0.03 in both frequencies. Then, the band $|\theta_y| \leqslant \frac{\pi}{A}, |\theta_x| \leqslant \pi$ is searched again with the θ_y-increment reduced to $0.03/A$; the θ_x increment is kept as 0.03.

As a remark on the multigrid results tabulated in Section 5.2, an inconsistent scheme does not necessarily lead to poor multigrid performance. Inconsistency does imply that the coarse-grid correction for the smoothest components is not precise. For example, LFA analyses show that multigrid convergence on isotropic ($A = 1$) triangular grids with coarse grids discretized with either of the two inconsistent ETO schemes is similar to multigrid convergence on isotropic quadrilateral grids. For high-aspect-ratio triangular grids, Table 11 indicates that, with the ETO (EN) scheme, the low-frequency coarse-grid correction is 5/6 of the optimal correction and the overall multigrid cycle is 0.2 per cycle. For high-aspect-ratio triangular grids with the ETO (FT) scheme, the coarse-grid correction for intermediate frequencies in x and low frequencies in the y-direction is inadequate, leading to poor multigrid convergence.

Table 11
Expansion of coarse-grid discretization operators on a triangular fine grid.

Scheme	$A = 1$ $-h_y^2\widehat{L}_H''$	$\lim A \to \infty$ $-h_y^2\widehat{L}_H''$
Avg-LSQ	$\theta_x^2+\theta_y^2$	θ_y^2
ETO (EN)	$\theta_x^2+\theta_y^2+\frac{1}{3}\theta_x\theta_y$	$\frac{5}{6}\theta_y^2$
ETO (FT)	$\frac{31}{30}\theta_x^2+\frac{31}{30}\theta_y^2+\frac{1}{3}\theta_x\theta_y$	$\frac{11}{12}\theta_y^2+\frac{7}{60}\theta_x^2+\frac{1}{6}\theta_x\theta_y$

The cause of the slowdown is the increase of the stencil weights in the x-direction associated with skew angles approaching $\pi/2$. The same difficulty occurs for the Avg-LSQ (FT) scheme, even though it is a consistent scheme.

Appendix C. Idealized relaxation on high-aspect-ratio grids

The effects of various idealized and actual relaxation schemes on multigrid convergence are shown below for one coarse-grid discretization—the Avg-LSQ (EN) scheme. Regular triangular and quadrilateral grids are considered, following the groundwork in Appendix B.

Table 12 shows convergence rates of 2-grid cycles computed with LFA for quadrilateral and triangular fine grids with $A = 10^4$. The results are shown with residual averaging although the conclusions are not sensitive to its inclusion. Four ideal relaxations, IR-P and IR-L with multicolor and lexicographic ordering, and two actual relaxations, line-implicit with 2-color and lexicographic ordering, are considered. The actual line-implicit relaxations are less than 0.12 per cycle for both triangular and quadrilateral grids.

The 4-color (1234) IR-P cycle is unstable and thus not suitable as a predictor of the actual cycle. Although not shown, other color sequences give similar results. Convergence of the lexicographic IR-P cycle is better than 0.1 per cycle and thus lexicographic IR-P could be considered as a possible idealized relaxation. However, the IR-L cycles are uniformly-excellent quantitative predictors when the idealized relaxation is applied in the same order as the actual line-implicit relaxation. Convergence of the 2-color IR-L cycle predicts convergence of the actual cycle with 2-color line-implicit relaxation. Likewise, convergence of the lexicographic IR-L cycle predicts convergence of the actual cycle with lexicographic line-implicit relaxation. The IR-L cycle is a simple, consistent, and accurate predictor of the convergence rates of the actual cycle, and we use it for the analyses of multigrid solutions on high-aspect-ratio grids reported in Section 5.2.

Note that the instability of the IR-P cycle occurs for error components that are extremely smooth in the y-direction. It is difficult to observe this instability in actual computations because, to realize such smooth components, a large number of high-aspect-ratio cells in the y-direction is required. Table 13 illustrates this behavior, showing convergence rates computed with LFA and confirmed in actual computations on uniform high-aspect-ratio grids in a

Table 12
LFA convergence rates per cycle for triangular and quadrilateral grids with residual averaging; actual is line-implicit relaxation.

Relaxation	Order	Quadrilateral fine grid	Triangular fine grid
IR-P	4-Color (1234)	46	152
IR-P	Lexicographic	0.1	0.07
IR-L	2-Color	0.11	0.12
IR-L	Lexicographic	0.06	0.07
Actual	2-Color	0.11	0.07
Actual	Lexicographic	0.02	0.07

Table 13
Convergence rates of 4-color (1234) IR-P(2,2) cycles as a function of grid size and aspect ratio for periodic domains with residual averaging; quadrilateral fine grid.

Fine grid	Convergence rate	
	$A = 1$	$A = 10^4$
32^2	0.109	0.109
128^2	0.110	0.193
512^2	0.110	0.626
2048^2	0.110	2.369
∞	0.110	46

Author's personal copy

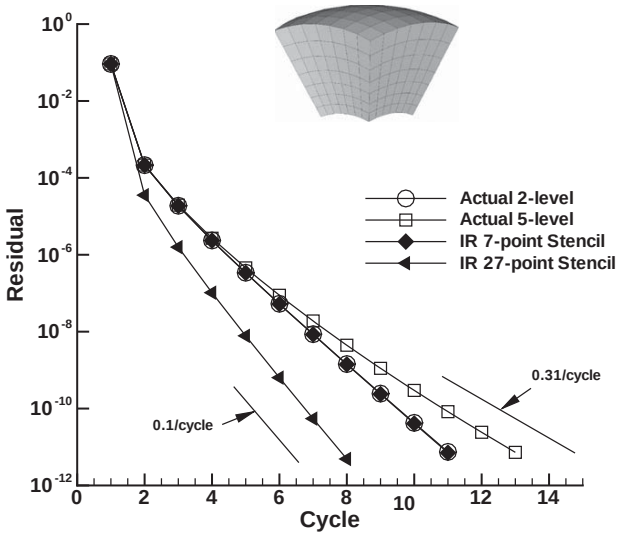


Fig. 16. Convergence rates of actual and IR cycles for 64^3 hexahedral grid using point-wise relaxation; inset is a coarser $8 \times 8 \times 8$ grid.

periodic domain. The convergence rates are for 4-color (1234) IR-P(2,2) cycles with residual averaging on quadrilateral grids with $A = 1$ and $A = 10^4$; only Fourier components realizable on the specified grids have been considered. Grid-independent convergence is shown on isotropic grids ($A = 1$) but the instabilities on anisotropic grids ($A = 10^4$) have not reached their asymptotic value (from Table 12) for the entries in the table corresponding to 2048^2 points.

Appendix D. Idealized relaxation on isotropic grids

Here we show a somewhat subtle effect that arises in unstructured grids with IR based on edge-connections. The role effected by IR depends on the number of edges N_e in (20). For a hexahedral mesh, the number of simply-connected edges is 6 but the total number of simply-connected and virtual edges is 26, corresponding to 7-point and 27-point stencils of A^{IR}, respectively. The convergence of 4-color IR-P(2,2) for a 64^3 isotropic hexahedral grid over a spherical domain is shown in Fig. 16 for these two stencils. With the 27-point stencil, the asymptotic convergence of IR-P is noticeably faster than that with the 7-point stencil. Although not shown, even with single grid (no multigrid) iterations, relaxation of (20) with the 27-point stencil converges in half of the iterations as that with the 7-point stencil.

On this particular grid, the actual discrete diffusion operator is much closer to the 7-point operator. As seen in Fig. 16, convergence of actual 2-level V(2,2) cycles is quite close to that of IR with the 7-point stencil. Convergence of actual cycles with 5 levels is somewhat slower asymptotically than the 2-level convergence. The interpretation is that IR with the 27-point stencil is providing faster convergence of the medium frequencies than point relaxation of the actual diffusion operator. Using additional relaxation provides convergence rates per cycle that agree closely to that with the 27-point stencil but does not provide an overall gain in efficiency.

References

[1] Barth TJ. Numerical aspects of computing high-Reynolds number flow on unstructured meshes. AIAA paper 91-0721, January 1991.
[2] Diskin B, Thomas JL, Mineck R. On quantitative analysis methods for multigrid solutions. SIAM J Sci Comput 2005;27(1):108–29.
[3] Diskin B, Thomas JL. Accuracy analysis for mixed-element finite-volume discretization schemes. NIA technical report 2007-8, August 2007.
[4] Diskin B, Thomas JL, Nielsen EJ, Nishikawa H, White JA. Comparison of node-centered and cell-centered unstructured finite-volume discretizations: viscous fluxes. AIAA paper 2009-597. AIAA J, 2010;48(7).
[5] Diskin B, Thomas JL. Notes on accuracy of finite-volume discretization schemes on irregular grids. Appl Numer Math. 2010;60:224–60.
[6] Brandt A. Multi-level adaptive solutions to boundary-value problems. Math Comput 1977;31:333–90.
[7] Haselbacher AC. A grid-transparent numerical method for compressible viscous flow on mixed unstructured meshes. PhD thesis, Loughborough University; 1999.
[8] Hyams DG, Sreenivas K, Whitfield DL. Parallel FAS multigrid for arbitrary Mach number, high Reynolds number unstructured flow solvers. AIAA paper 2006-2821, June 2006.
[9] Mavriplis DJ. Multigrid techniques for unstructured meshes. VKI lecture series VKI-LS 1995-02. Belgium: Von Karman Institute for Fluid Dynamics, Rhode-Saint-Genese; 1995.
[10] Mavriplis DJ. Unstructured grid techniques. Annu Rev Fluid Mech 1997;29:473–514.
[11] Mavriplis DJ, Pirzadeh S. Large-scale parallel unstructured mesh computations for 3d high-lift analysis. J Aircraft 1999;36(6):987–98.
[12] Mavriplis DJ. An assessment of linear versus non-linear multigrid methods for unstructured mesh solvers. J Comput Phys 2002;175:302–25.
[13] Nishikawa H, Diskin B, Thomas JL. Critical study of agglomerated multigrid methods for diffusion. AIAA paper 2009-4138. AIAA J. 2010;48(4).
[14] Svärd M, Nordström J. An accuracy evaluation of unstructured node-centered finite-volume methods. Appl Numer Math 2008;58(8):1142–58. also available as NIA report 2005-04. NASA CR-2006-214293, April 2006.
[15] Thomas JL, Diskin B, Brandt A. Textbook multigrid efficiency for the incompressible Navier–Stokes equations: high Reynolds number wakes and boundary layers. Comput Fluids 2001;30:853–74.
[16] Thomas JL, Diskin B, Brandt A. Textbook multigrid efficiency for fluid simulations. Annu Rev Fluid Mech 2003;35:317–40.
[17] Thomas JL, Diskin B, Rumsey CL. Towards verification of unstructured grid methods. AIAA J 2008;46(12):3070–9. also AIAA paper 2008-0666.
[18] Trottenberg U, Oosterlee CW, Schüler A. Multigrid. London: Academic Press; 2000.
[19] Venkatakrishnan V. Perspective on unstructured grid solvers. AIAA J 1996;34(3):533–45.

20th AIAA Computational Fluid Dynamics Conference
27 - 30 June 2011, Honolulu, Hawaii

AIAA 2011-3232

Development and Application of Parallel Agglomerated Multigrid Methods for Complex Geometries

Hiroaki Nishikawa* and Boris Diskin†
National Institute of Aerospace, Hampton, VA 23666

We extend previous serial developments of agglomerated multigrid techniques for fully unstructured grids in three dimensions to parallel computations. We demonstrate a robust parallel fully-coarsened agglomerated multigrid technique for the Euler, the Navier-Stokes, and the RANS equations for 3D complex geometries, incorporating the following key developments: consistent and stable coarse-grid discretizations, a hierarchical agglomeration scheme, and line-agglomeration/relaxation using prismatic-cell discretizations in the highly-stretched grid regions. A significant speed-up in computer time over state-of-art large-scale computations is demonstrated.

I. Introduction

Multigrid techniques [1] are used to accelerate convergence of current Reynolds-Averaged Navier-Stokes (RANS) solvers for both steady and unsteady flow solutions, particularly for structured-grid applications. Mavriplis et al. [2, 3, 4, 5] pioneered agglomerated multigrid methods for large-scale unstructured-grid applications. However, systematic computations with these techniques showed a serious convergence degradation on highly-refined grids. To overcome the difficulty, we critically studied agglomerated multigrid techniques [6,7] for two- and three-dimensional isotropic and highly-stretched grids and developed quantitative analysis methods and computational techniques to achieve grid-independent convergence for a model diffusion equation representing laminar diffusion in the incompressible limit. It was found in Ref. [6] that it is essential for grid-independent convergence to use *consistent* coarse-grid discretizations. In the later Ref. [7], it was found that the use of *prismatic cells and line-agglomeration/relaxation* is essential for grid-independent convergence on fully-coarsened highly-stretched grids. Building upon these fundamental studies, we extended and demonstrated these techniques for a model diffusion, inviscid, and Reynolds-Averaged Navier-Stokes (RANS) equations over complex geometries using a serial code in Ref. [8]. In this paper, we present a parallel version of the agglomerated multigrid code.

The paper is organized as follows. Finite-volume discretizations employed for target grids are described. Details of the hierarchical agglomeration scheme are described with a particular parallel implementation. Elements of the multigrid algorithm are then described, including discretizations on coarse grids. Multigrid results for complex geometries are shown for the Euler, the Navier-Stokes, and the RANS equations. The final section contains conclusions.

II. Discretization

The discretization method is a finite-volume discretization (FVD) centered at nodes. It is based on the integral form of governing equations of interest:

$$\oint_{\Gamma} (\mathcal{F} \cdot \hat{\mathbf{n}})\, d\Gamma = \iint_{\Omega} s\, d\Omega, \tag{1}$$

where $\mathcal{F}$ is a flux tensor, s is a source term, Ω is a control volume with boundary Γ, and $\hat{\mathbf{n}}$ is the outward unit normal vector. The governing equations are the Euler equations, the Navier-Stokes equations, and the

*Senior Research Scientist (hiro@nianet.org), National Institute of Aerospace, 100 Exploration Way, Hampton, VA 23666 USA, Member AIAA

†Associate Fellow (bdiskin@nianet.org), National Institute of Aerospace, 100 Exploration Way, Hampton, VA 23666 USA.; Visiting Associate Professor, Mechanical and Aerospace Engineering Department, University of Virginia, Charlottesville, Senior Member AIAA

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

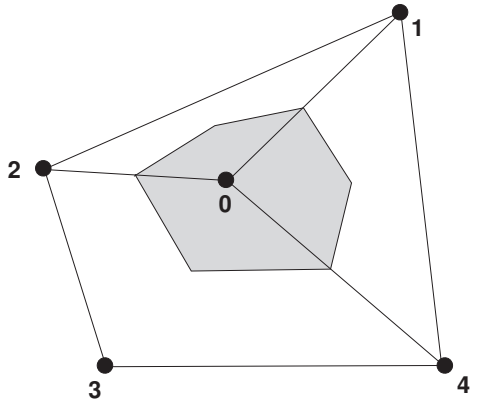


Figure 1. Illustration of a node-centered median-dual control volume (shaded). Dual faces connect edge midpoints with primal cell centroids. Numbers 0-4 denote grid nodes.

RANS equations with the Spalart-Allmaras one-equation model [9]. For inviscid flow problems, the governing equations are the Euler equations. Boundary conditions are a slip-wall condition and inflow/outflow condition. For viscous flow problems, boundary conditions are non-slip conditions on walls and inflow/outflow conditions on open boundaries. The source term, s, is zero except for the turbulence-model equation (see Ref. [9]).

The general FVD approach requires partitioning the domain into a set of non-overlapping control volumes and numerically implementing Equation (1) over each control volume. Node-centered schemes define solution values at the mesh nodes. In 3D, the primal cells are tetrahedra, prisms, hexahedra, or pyramids. The *median-dual* partition [10, 11] used to generate control volumes is illustrated in Figure 1 for 2D. These non-overlapping control volumes cover the entire computational domain and compose a mesh that is dual to the primal mesh.

The main target discretization of interest for the viscous terms of the Navier-Stokes and RANS equations is obtained by the Green-Gauss scheme [12, 13], which is a widely-used viscous discretization for node-centered schemes and is equivalent to a Galerkin finite-element discretization for tetrahedral grids. For mixed-element cells, edge-based contributions are used to increase the h-ellipticity of the operator [12, 13]. This augmentation is done by the face-tangent construction [7] with the efficient implementation that is independent of the face-tangent vectors (see Appendices of Ref. [14]); thus the resulting scheme is called here the face-tangent Green-Gauss scheme. The inviscid terms are discretized by a standard edge-based method with unweighted least-squares gradient reconstruction and Roe's approximate Riemann solver [15, 16]. Limiters are not used for the problems considered in this paper. The convection terms of the turbulence equation are discretized with first-order accuracy.

III. Agglomeration Scheme

III.A. Hierarchical Agglomeration Scheme

As described in the previous papers [6,7,8], the grids are agglomerated within a topology-preserving framework, in which hierarchies are assigned based on connections to the computational boundaries. *Corners* are identified as grid points with three or more boundary-condition-type closures (or three or more boundary slope discontinuities). *Ridges* are identified as grid points with two boundary-condition-type closures (or two boundary slope discontinuities). *Valleys* are identified as grid points with a single boundary-condition-type closure. *Interiors* are identified as grid points without any boundary condition. The agglomerations proceed hierarchically from seeds within the topologies — first corners, then ridges, then valleys, and finally interiors. Rules are enforced to maintain the boundary condition types of the finer grid within the agglomerated grid. Candidate volumes to be agglomerated are vetted against the hierarchy of the currently agglomerated volumes. As in the previous work, we use the rules summarized in Table 1. In order to enable a valid non-degenerate stencil for linear prolongation and least-squares gradients near boundaries [7], the rules reflect less agglomerations near boundaries than in the interior. Corners are never agglomerated, ridges are agglomerated only with ridges, and valleys are agglomerated only with valleys. A typical boundary agglomeration generated by the above rules is shown in Figure 2. The conditional entries denote that further inspection of the connectivity of the topology must be considered before agglomeration is allowed. For example, a ridge can be agglomerated into an existing ridge agglomeration

Hierarchy of Agglomeration	Hierarchy of Added Volume	Agglomeration Admissibility
corner	any	disallowed
ridge	interior	disallowed
ridge	valley	disallowed
ridge	ridge	conditional
valley	interior	disallowed
valley	valley	conditional
interior	interior	allowed

Table 1. Admissible agglomerations.

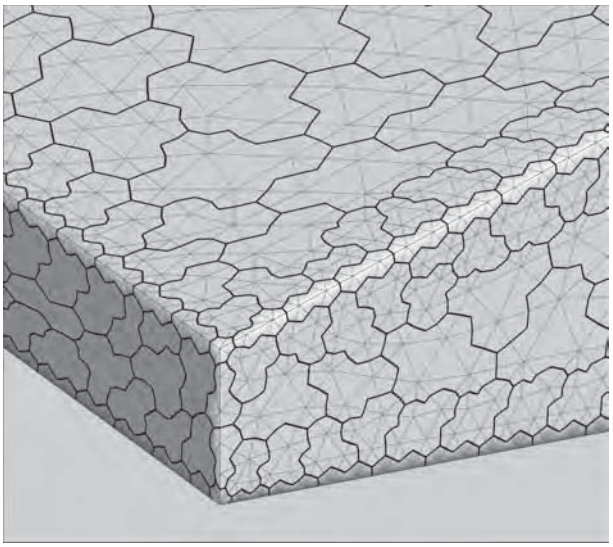


Figure 2. Trailing-edge area of a 3D wing agglomerated by the hierarchical scheme. Primal grid is shown by thin lines; agglomerated grid is shown by thick lines.

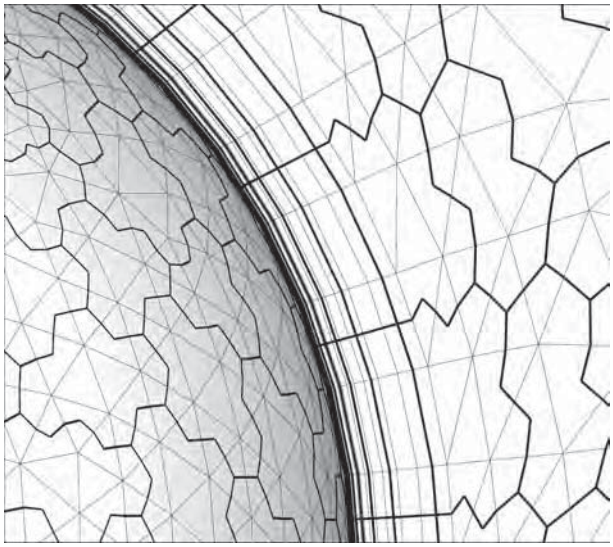


Figure 3. Typical implicit line-agglomeration showing a curved solid body surface on the left and a symmetry plane on the right. The projection of the line-agglomerations can be seen on the symmetry plane.

if the two boundary conditions associated with each ridge are the same. For valleys or interiors, all available neighbors are collected and then agglomerated one by one in the order of larger number of edge-connections to a current agglomeration until the maximum threshold of agglomerated nodes (4 for valleys; 8 for interiors) is reached. The prolongation operator P_1 is modified to prolong only from hierarchies equal or above the hierarchy of the prolonged point. Hierarchies on each agglomerated grid are inherited from the finer grid.

As in the previous work [8], we perform the agglomeration in the following sequence:

1. Agglomerate viscous boundaries (bottom of implicit lines).
2. Agglomerate prismatic layers through the implicit lines (implicit-line agglomeration).
3. Agglomerate the rest of the boundaries.
4. Agglomerate the interior.

The second step is a line-agglomeration step where volumes are agglomerated along implicit lines starting from the volume directly above the boundary volume. Specifically, we first agglomerate volumes corresponding to the second and third entries in the implicit-line lists associated with each of the fine-grid volumes contained in a boundary agglomerate. The line agglomeration continues to the end of the shortest line among the lines associated with the boundary agglomerate. This line-agglomeration process preserves the boundary agglomerates. Figure 3 illustrates typical implicit line-agglomeration near a curved solid body. The implicit line-agglomeration preserves the line structure of the fine grid on coarse grids, so that line-relaxations can be performed on all grids

to address the grid anisotropy. If no implicit lines are defined, typical for inviscid grids, the first two steps are skipped.

In each boundary agglomeration (steps 1 and 3), agglomeration begins with corners (ridges or valleys if corners do not exist), creates a front list defined by collecting volumes adjacent to the agglomerated corners, and proceeds to agglomerate volumes in the list (while updating the list as agglomeration proceeds) in the order of ridges and valleys. During the process, a volume is selected from among those in the same hierarchy that has the least number of non-agglomerated neighbors, thereby reducing the occurrences of agglomerations with small numbers of volumes. A heap data-structure is utilized to efficiently select such a volume. The agglomeration continues until the front list becomes empty. Finally, for both valleys and interiors, agglomerations containing only a few volumes (typically one) are combined with other agglomerations.

III.B. Parallel Implementation

For parallel implementation, the hierarchical agglomeration algorithm is applied independently to each partition. That is, no agglomeration is performed across partition boundaries. In each partition, we first select a starting volume in the priority order: corner, ridge, valley, and interior. Then, we execute the hierarchical agglomeration described above within the partition. In rare cases, a partition consists of a few disjoint grids. If such a partition is found, e.g., by a neighbor-to-neighbor search, we set up a starting volume in each disjoint grid to fully agglomerate the partition. No special modification is necessary for the line agglomeration as our partitioning guarantees that all nodes in each implicit-line belong to the same partition. Due to the advancing-front nature of the agglomeration scheme, the resulting agglomerated grids will be different for different numbers of partitions. However, no significant dependence is observed in the numerical results presented in this paper. Partition-independent agglomeration remains a challenge; it is a subject of future work.

IV. Single-grid Iterations

The single-grid iteration scheme is based on the implicit formulation:

$$\left(\frac{\Omega}{\Delta\tau} + \frac{\partial \hat{R}^*}{\partial U}\right)\delta U = -\hat{R}(U), \tag{2}$$

where $\hat{R}(U)$ is the target residual computed for the current solution U, $\Delta\tau$ is a pseudo-time step, $\frac{\partial \hat{R}^*}{\partial U}$ is an exact/approximate Jacobian, and δU is the change to be applied to the solution U. An approximate solution to Equation (2) is computed by a certain number of relaxations on the linear system (linear-sweeps). Update of U completes one nonlinear iteration. The RANS equations are iterated in a loosely-coupled formulation: first the mean-flow variables are updated, and then the turbulence residual is evaluated and the turbulence variable is updated. The left-hand-side operator of Equation (2) includes the exact linearization of the viscous terms and a linearization of the inviscid terms involving first-order contributions only. Thus, the iterations represent a variant of defect correction. Typically in our single-grid RANS applications, the first-order Jacobian corresponds to the linearization of Van Leer's flux-vector splitting [17]. But the linearization of Roe's approximate Riemann solver is also available. In this study, Jacobians are updated after each iteration.

The linear sweeps performed before each nonlinear update include ν_p sweeps of the point multi-color Gauss-Seidel relaxation performed through the entire domain followed by ν_l line-implicit sweeps in stretched regions. The line-implicit sweeps are applied only when solving the Navier-Stokes or RANS equations. In a line-implicit sweep, unknowns associated with each line are swept simultaneously by inverting a block tridiagonal matrix [7]. Our single-grid computations do not represent the default FUN3D usage; they differ in the Jacobian type and update strategy and the use of implicit-lines.

V. Multigrid

V.A. Multigrid V-Cycle

The multigrid method is based on the full-approximation scheme (FAS) [1, 18] where a coarse-grid problem is solved/relaxed for the solution approximation. A correction, computed as the difference between the restricted fine-grid solution and the coarse-grid solution, is prolonged to the finer grid to update the fine-grid solution. The two-grid FAS is applied recursively through increasingly coarser grids to define a V-cycle. A V-cycle, denoted as $V(\nu_1, \nu_2)$, uses ν_1 relaxations performed at each grid before proceeding to the coarser grid and ν_2 relaxations

after coarse-grid correction. On the coarsest grid, relaxations are performed to bring two orders of magnitude residual reduction or until the maximum number of relaxations, 10, is reached.

V.B. Inter-Grid Operators

The control volumes of each agglomerated grid are found by summing control volumes of a finer grid. An operator that performs the summation is given by a conservative agglomeration operator, R_0, which acts on fine-grid control volumes and maps them onto the corresponding coarse-grid control-volumes. Any agglomerated grid can be defined, therefore, in terms of R_0 as

$$\Omega^c = R_0 \Omega^f, \tag{3}$$

where superscripts c and f denote entities on coarser and finer grids, respectively. On the agglomerated grids, the control volumes become geometrically more complex than their primal counterparts and the details of the control-volume boundaries are not retained. The directed area of a coarse-grid face separating two agglomerated control volumes, if required, is found by lumping the directed areas of the corresponding finer-grid faces and is assigned to the virtual edge connecting the centers of the agglomerated control volumes.

Residuals on the fine grid, $\hat{R}^f$, corresponding to the integral equation (1), are restricted to the coarse grid by the conservative agglomeration operator, R_0, as

$$\hat{R}^c = R_0 \hat{R}^f, \tag{4}$$

where $\hat{R}^c$ denotes the fine-grid residual restricted to the coarse grid. The fine-grid solution approximation, U^f, is restricted as

$$U_0^c = \frac{R_0(U^f \Omega^f)}{\Omega^c}, \tag{5}$$

where U_0^c denotes the fine-grid solution approximation restricted to the coarse grid. The restricted approximation is then used to define the forcing term to the coarse-grid problem as well as to compute the correction, $(\delta U)^c$:

$$(\delta U)^c = U^c - U_0^c, \tag{6}$$

where U^c is an updated coarse-grid solution obtained directly from the coarse-grid problem. The correction to the finer grid is prolonged typically through the prolongation operator, P_1, that is exact for linear functions, as

$$(\delta U)^f = P_1 (\delta U)^c. \tag{7}$$

The operator P_1 is constructed locally using linear interpolation from a tetrahedra defined on the coarse grid. The geometrical shape is anchored at the coarser-grid location of the agglomerate that contains the given finer control volume. Other nearby points are found by the adjacency graph. An enclosing simplex is sought that avoids prolongation with non-convex weights and, in situations where multiple geometrical shapes are found, the first one encountered is used. Where no enclosing simplex is found, the simplex with minimal non-convex weights is used.

V.C. Coarse-Grid Discretizations

For inviscid coarse-grid discretization, a first-order edge-based scheme is employed. For the viscous term, two classes of coarse-grid discretizations were previously studied [6, 7]: the Average-Least-Squares (Avg-LSQ) and the edge-terms-only (ETO) schemes. The consistent Avg-LSQ schemes are constructed in two steps: first, LSQ gradients are computed at the control volumes; then, the average of the control-volume LSQ gradients is used to approximate a gradient at the face, which is augmented with the edge-based directional contribution to determine the gradient used in the flux. There are two variants of the Avg-LSQ scheme. One uses the average-least-squares gradients in the direction normal to the edge (edge-normal gradient construction). The other uses the average-least-squares gradients along the face (face-tangent gradient construction [7]).

The ETO discretizations are obtained from the Avg-LSQ schemes by taking the limit of zero Avg-LSQ gradients. The ETO schemes are often cited as a thin-layer discretization in the literature [2, 3, 4]; they are positive schemes but are not consistent (i.e., the discrete solutions do not converge to the exact continuous solution with consistent grid refinement) unless the grid is orthogonal [16, 19]. As shown in the previous papers [6, 7], ETO schemes lead to deterioration of the multigrid convergence for refined grids, and therefore are not considered in this paper. For practical applications, the face-tangent Avg-LSQ scheme was found to

	Inviscid	Viscous (Diffusion)
Primal grid	Second-order edge-based reconstruction	Face-Tangent Green-Gauss
Coarse grids	First-order edge-based reconstruction	Face-Tangent Avg-LSQ

Table 2. Summary of discretizations used to define the residual, $\hat{R}$.

	Inviscid	Viscous
Primal grid	Approximate (first-order scheme)	Exact ($\hat{R}^* = \hat{R}$)
Coarse grids	Exact or Approximate	Approximate (edge-terms only)

Table 3. Summary of Jacobians, $\frac{\partial \hat{R}^*}{\partial U}$.

be more robust than the edge-normal Avg-LSQ scheme [8]. It provides superior diagonal dominance in the resulting discretization [6, 7]. In this study, we employ the face-tangent Avg-LSQ scheme [7] as a coarse-grid discretization of the viscous term. It has been implemented in the form independent of the face-tangent vectors (see Appendices of Ref. [14]). For excessively-skewed faces over 90° angle between the outward face normal and the corresponding outward edge vector, which can arise on agglomerated grids, the viscous fluxes are ignored. For inviscid discretization, we employ a first-order edge-based discretization on coarse grids. Table 2 shows a summary of discretizations used.

V.D. Relaxations

The relaxation scheme is similar to the single-grid iteration described in Section IV with the following important differences. On coarse grids, the Avg-LSQ scheme used for viscous terms has a larger stencil than the Green-Gauss scheme implemented on the target grid and its exact linearization has not been used; instead, an approximate linearization based on the corresponding ETO scheme is used. For the inviscid part, the first-order Jacobian is constructed based on Van Leer's flux-vector splitting or Roe's approximate Riemann solver in accordance with the linearization employed on the target grid. If the latter is employed, the linearization will be exact on coarse grids where the first-order scheme is used for the residual.

Table 3 summarizes the Jacobians used for inviscid and viscous terms on the primal and coarse grids. The Jacobians are updated in all levels at the beginning of a cycle and frozen through the end of the cycle. Compared with the single-grid scheme in which the Jacobians are updated at every iteration in this study, this strategy saves a significant amount of computing time for multigrid. As in the previous work [8], significantly fewer linear sweeps are used in a multigrid relaxation than in a single-grid iteration: typically, $\nu_p = \nu_l = 5$ for both the mean flow and turbulence relaxations.

VI. Numerical Results

VI.A. Inviscid Flows

The multigrid method was applied to two inviscid cases: a wing-body configuration (1,012,189 nodes), and a wing-flap configuration (1,184,650 nodes). The inflow Mach number is 0.3, the angles of attack are 0.0 for the wing-body configuration and 2.0 degrees for the wing-flap configuration. The multigrid $V(2,1)$ cycle of 3 levels was employed for these cases, with 4, 8, 12, and 16 processors. For these inviscid cases, the full-multigrid algorithm was employed to obtain the initial solution on the target grid. Also, the relaxation is based on a linearization of the Roe flux in the multigrid. For linear sweeps, we set $(\nu_p, \nu_l) = (15, 0)$ for the single-grid scheme, and $(\nu_p, \nu_l) = (5, 0)$ for the multigrid.

The CFL number is ramped from 10 to 200 during the first 10 iterations/cycles for single-grid/multigrid calculations. All cases have been run until the residual reaches the machine zero, 10^{-15}.

Figure 4 shows grids and results for the wing-body configuration case. The convergence results for all processors are given in Figure 4(d); it shows that the convergence is nearly independent of the number of processors (i.e., the multigrid lines are overlapped, and so are the single-grid lines). Figure 4(e) shows the convergence results for 16 processors. It shows that the multigrid converges 5 times faster in CPU time than

	Wing-Body(0.2M)	Wing-Body(1.0M)	Wing-Flap(1.2M)
Multigrid V(2,1)	0.400(34)	0.530(47)	0.860(66)
Single Grid	0.955(370)	0.958(680)	0.956(459)

Table 4. Asymptotic convergence rates for the inviscid case. Numbers in the parenthesis are single-grid iterations or multigrid cycles to convergence.

	0.6M	1.4M	2.7M	4.5M
Multigrid V(3,3)	0.789(68)	0.854(107)	0.826(93)	0.820(118)
Single Grid	0.944(479)	0.967(647)	0.980(907)	1.00

Table 5. Asymptotic convergence rates for the laminar case. Numbers in the parenthesis are single-grid iterations or multigrid cycles to convergence.

the single-grid relaxations. A reasonable parallel scalability can be observed in Figure 4(f) where the solid lines indicate the perfect scaling. It shows also that the speed-up factor is almost independent of the number of processors. Figure 5 shows convergence results for the same wing-body configuration case with two different sizes of grids: 0.2 and 1 million grids. As can be seen in Figure 5(a), the multigrid convergence is not exactly grid-independent, but the dependency is much weaker than the single-grid convergence dependence. Translated into the CPU time, it implies a substantial speed-up for larger-scale problems. Figure 5(b) shows in fact that the multigrid is about 2 times faster then the single-grid scheme for the 0.2-million grid, and 5 times faster for the one-million grid.

Figure 6 shows grids and results for the wing-flap configuration case. The processor-independent convergence is demonstrated in Figures 6(d). Figure 6(e) shows that the multigrid converges nearly 3 times faster in CPU time than the single-grid scheme. A reasonable parallel scalability is demonstrated in Figure 6(f).

In both inviscid cases, the cost of one multigrid $V(2,1)$ cycle is roughly equal to three single-grid iterations. Typically, one would expect that one multigrid $V(2,1)$ cycle is equivalent to 4 single-grid iterations. However, the multigrid requires a less number of linear-sweeps than the single-grid iteration, which can cut a significant portion of the expected cost. See Ref. [8] for a detailed cost comparison.

Asymptotic convergence rates are shown in Table 4, which are averaged rates over the last 10 cycles/iterations and over the four-different-processor cases.

VI.B. Laminar Flow

For viscous flow applications, we encountered a significant slow down in multigrid convergence, but then found that additional relaxations improve the performance. We thus applied the multigrid algorithm with 3-level $V(3,3)$ to a laminar flow over a hemisphere cylinder. The inflow Mach number is 0.2, the angle of attack is zero, and the Reynolds number is 400. We performed a convergence study using four different grids: 0.6, 1.4, 2.7 and 4.5 million nodes. Each grid is a mixed grid with a highly-stretched prismatic grid around the hemisphere cylinder and isotropic tetrahedra elsewhere. The line-agglomeration/relaxation algorithm was applied in the stretched region. For both multigrid and single-grid calculations, the CFL number is 200 and the linearization of Roe's approximate Riemann solver was used as a driver. The CFL number is ramped from 10 to 200 during the first 500 iterations for the single-grid calculations and 50 cycles for the multigrid calculations. The number of linear point/line-sweeps is 25 for the single-grid calculations, and 10 for the multigrid calculations. The number of processors used here is 16. The use of the linearization of the Roe flux and a larger number of linear sweeps were necessary for both schemes to converge in all cases although the single-grid scheme still fails to converge for the finest grid.

Figure 7 shows grids and convergence results. Figure 7(d) shows the convergence results. The single-grid scheme shows a consistent increase in the number of iterations with the number of nodes. It also shows that it is non-convergent for the finest grid. On the other hand, the multigrid converged on all grids. Results in Figure 7(d) indicate that the number of cycles to convergence varies slightly with the number of nodes, implying the grid-independent convergence of the multigrid (see Table 5 for the number of cycles). In terms of CPU time, Figure 7(e) shows that the multigrid is nearly four times faster then the single-grid scheme on the grid of 2.7M nodes. Table 5 summarizes the asymptotic convergence rates (averaged over the last 50 cycles/iterations) observed in the numerical results. It shows that the convergence rate (per iteration) for the single grid scheme

	Mean flow	Turbulence
Multigrid V(3,3)	0.940	0.913
Single Grid	0.980	0.996

Table 6. Asymptotic convergence rates for the RANS case.

deteriorates as the grid gets finer whereas the convergence rate (per cycle) for the multigrid does not deteriorate as the grid gets finer. Finally, these results indicate that the cost of one multigrid $V(3,3)$ cycle is roughly equal to two single-grid iterations.

VI.C. Turbulent Flows (RANS)

We applied the 3-level $V(3,3)$ multigrid algorithm to a RANS simulation on the DPW-W2 grid (1.88 million nodes), with 16, 20, and 36 processors. The inflow Mach number is 0.76, the angle of attack is 0.5 degree, and the Reynolds number is 5 million. The initial solution is a free stream condition. For the single-grid scheme, the CFL number is ramped from 10 to 200 for the mean-flow equations and and 1 to 30 for turbulence equation over the first 50 iterations. For the multigrid scheme, the CFL number is ramped from 10 to 500 for the mean-flow equations and 10 to 300 for turbulence equation over the first 50 cycles. The grid is, again, a mixed grid with an isotropic tetrahedral region and a highly-stretched prismatic layer around the wing. As in the laminar case, the line-agglomeration/relaxation algorithm was applied in the stretched region. In both multigrid and single-grid calculations, the linearization of Van Leer's flux-vector splitting scheme was used as a driver. The number of linear point/line-sweeps is 15 for the mean-flow equations and 10 for the turbulence equation in the single-grid calculations. For the multigrid calculations, it is 5 for the mean-flow and turbulence equations.

Grids and results are shown in Figures 8 and 9. The convergence results for all processors are given in Figure 8(d); it shows that the convergence is nearly independent of the number of processors. Figure 8(e) shows the convergence results for 16 processors; it shows that the multigrid converges about three times as fast in CPU time as the single-grid scheme. The parallel scalability is consistent with the single-grid scheme, as can be observed in Figure 8(f). Figures 9(a) and 9(b) show the convergence results for the turbulence equation. Again, the convergence is nearly independent of the number of processors. The speed-up factor in CPU time is nearly 7 for the turbulence equation. Asymptotic convergence rates, obtained as averaged rates over the last 50 cycles/iterations and over the three-different-processor cases, are given in Table 6. For this problem, the multigrid converged in 160 cycles while the single grid scheme converged in 1958 iterations. These results indicate that similarly to the laminar case, the cost of one multigrid $V(3,3)$ cycle is roughly equal to two single-grid iterations. Finally, the lift and drag coefficients are 0.4865003979 and 0.020783234900, respectively, for all cases: they are identical up to 10 and 11 significant digits, respectively.

VII. Concluding Remarks

A parallel agglomerated multigrid algorithm has been developed and applied to inviscid and viscous flow problems over realistic geometries. A robust fully-coarsened hierarchical agglomeration scheme has been extended for parallel computations. The developed method was applied to the inviscid, laminar, and RANS simulations over realistic geometries. Numerical results show that impressive speed-ups can be achieved for realistic flow simulations. For the viscous cases, it was found that the relaxation scheme did not provide enough smoothing for the multigrid to work effectively and the use of $V(3,3)$ (instead of $V(2,1)$) greatly improved the multigrid convergence. For the laminar case, we have demonstrated that the multigrid method can achieve the grid-independent convergence. In future work, improvement in the viscous relaxation is desired on coarse grids. Future work includes also the implementation of the full multigrid algorithm for the RANS simulations, developing a rule to automatically determine the level of multigrid for given partitions, eliminating disjoint grids in a partition, etc. Eventually, the developed method will be applied to solve a wide range of larger-scale problems with more complex geometries. The grid-independent multigrid convergence will bring larger improvements over the single-grid convergence for larger-scale problems.

Acknowledgments

All results presented were computed within the FUN3D suite of codes at NASA Langley Research Center (`http://fun3d.larc.nasa.gov/`). The contributions of James L. Thomas and Dana P. Hammond at NASA Langley in the implementation within FUN3D are gratefully acknowledged. Also, we acknowledge the generous assistance of William L. Kleb (NASA Langley) in performing the RANS simulations. This work was supported by the NASA Fundamental Aeronautics Program through NASA Research Announcement Contract NNL07AA23C.

References

[1]Trottenberg, U., Oosterlee, C. W., and Schuller, A., *Multigrid*, Academic Press, 2000.

[2]Mavriplis, D. J., "Multigrid Techniques for Unstructured Meshes," *VKI Lecture Series VKI-LS 1995-02, Von Karman Institute for Fluid Dynamics, Rhode-Saint-Genese, Belgium*, 1995.

[3]Mavriplis, D. J., "Unstructured Grid Techniques," *Annual Review of Fluid Mechanics*, Vol. 29, 1997, pp. 473–514.

[4]Mavriplis, D. J. and Pirzadeh, S., "Large-Scale Parallel Unstructured Mesh Computations for 3D High-Lift Analysis," *Journal of Aircraft*, Vol. 36, No. 6, 1999, pp. 987–998.

[5]Mavriplis, D. J., "An Assessment of Linear versus Non-Linear Multigrid Methods for Unstructured Mesh Solvers," *Journal of Computational Physics*, Vol. 275, 2002, pp. 302–325.

[6]Nishikawa, H., Diskin, B., and Thomas, J. L., "Critical Study of Agglomerated Multigrid Methods for Diffusion," *AIAA Journal*, Vol. 48, No. 4, April 2010, pp. 839–847.

[7]Thomas, J. L., Diskin, B., and Nishikawa, H., "A Critical Study of Agglomerated Multigrid Methods for Diffusion on Highly-Stretched Grids," *Computers and Fluids*, Vol. 41, No. 1, February 2011, pp. 82–93.

[8]Nishikawa, H., Diskin, B., and Thomas, J. L., "Development and Application of Agglomerated Multigrid Methods for Complex Geometries," *40th AIAA Fluid Dynamics Conference and Exhibit*, AIAA Paper 2010-4731, Chicago, 2010.

[9]Spalart, P. R. and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA paper 92-0439, 1992.

[10]Barth, T. J., "Numerical Aspects of Computing Viscous High Reynolds Number Flows on Unstructured Meshes," AIAA Paper 91-0721, 1991.

[11]Haselbacher, A. C., *A Grid-Transparent Numerical Method for Compressible Viscous Flow on Mixed Unstructured Meshes*, Ph.D. thesis, Loughborough University, 1999.

[12]Anderson, W. K. and Bonhaus, D. L., "An implicit upwind algorithm for computing turbulent flows on unstructured grids," *Computers and Fluids*, Vol. 23, 1994, pp. 1–21.

[13]Diskin, B., Thomas, J. L., Nielsen, E. J., Nishikawa, H., and White, J. A., "Comparison of Node-Centered and Cell-Centered Unstructured Finite-Volume Discretizations: Viscous Fluxes," *AIAA Journal*, Vol. 48, No. 7, July 2010, pp. 1326–1338.

[14]Nishikawa, H., "Beyond Interface Gradient: A General Principle for Constructing Diffusion Schemes," *40th AIAA Fluid Dynamics Conference and Exhibit*, AIAA Paper 2010-5093, Chicago, 2010.

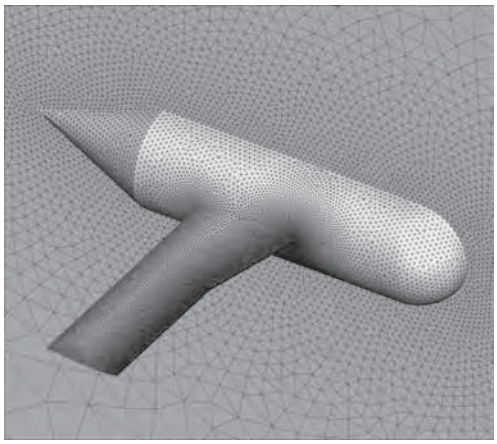
[15]Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, 1981, pp. 357–372.

[16]Diskin, B. and Thomas, J. L., "Accuracy Analysis for Mixed-Element Finite-Volume Discretization Schemes," *NIA Report No. 2007-08*, 2007.

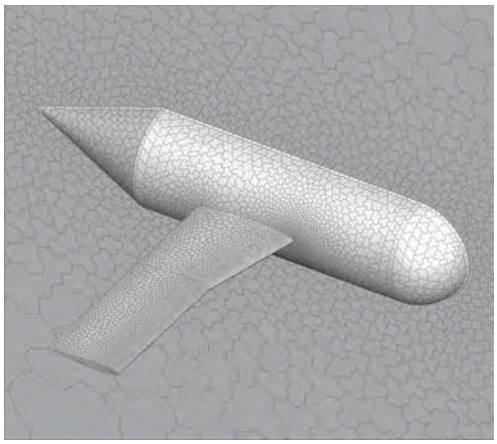
[17]van Leer, B., "Flux-vector splitting for the Euler equations," *Lecture Notes in Physics*, Vol. 170, Springer, 1982, pp. 507–512.

[18]Briggs, W. L., Henson, V. E., and McCormick, S. F., *A Multigrid Tutorial*, SIAM, 2nd ed., 2000.

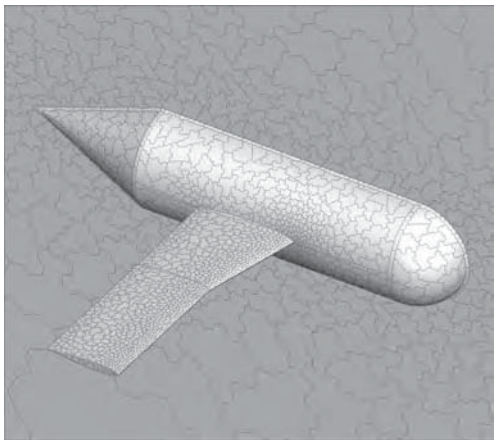
[19]Thomas, J. L., Diskin, B., and Rumsey, C. L., "Towards Verification of Unstructured Grid Methods," *AIAA Journal*, Vol. 46, No. 12, December 2008, pp. 3070–3079.



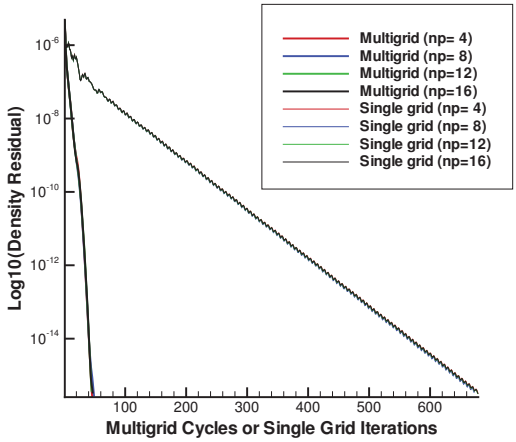
(a) Level 1: primal grid.



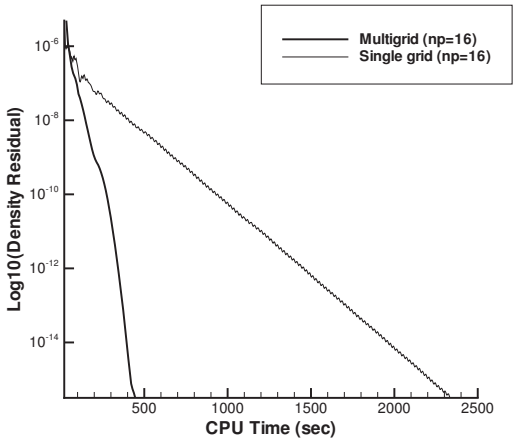
(b) Level 2: coarse grid.



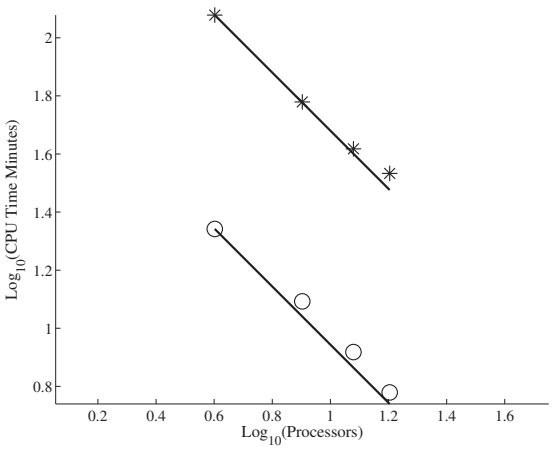
(c) Level 3: coarse grid.



(d) Convergence history (multigrid lines are overlapped; single-grid lines are overlapped.)

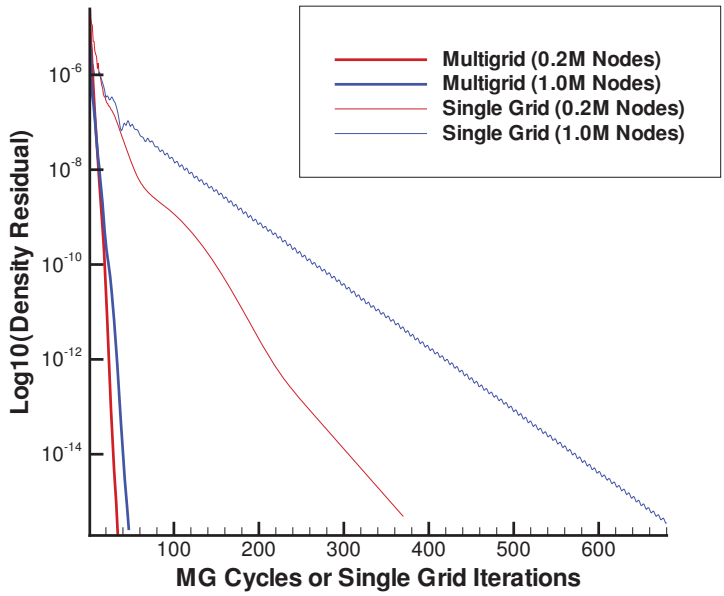


(e) Convergence history : residual vs. CPU time (16 processors)

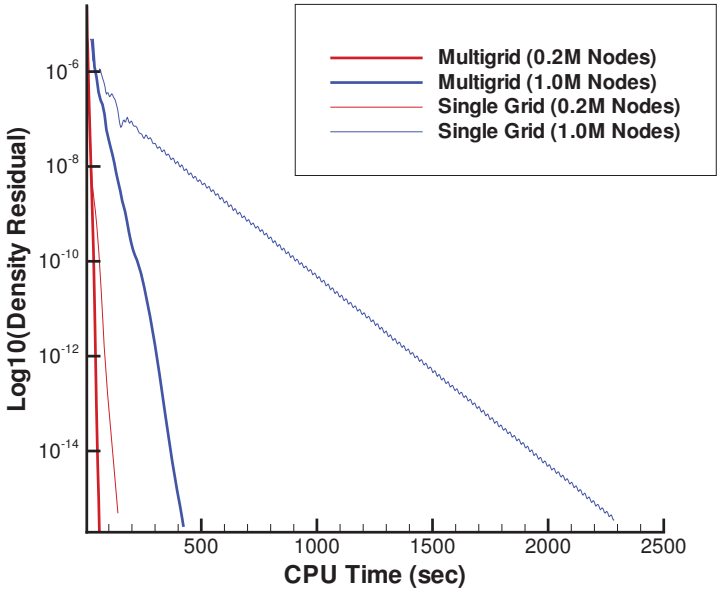


(f) Parallel Scalability (Stars for single grid; circles for multigrid; solid lines indicate the perfect scaling.)

Figure 4. Grids and convergence history for the wing-body inviscid case (1 million nodes).

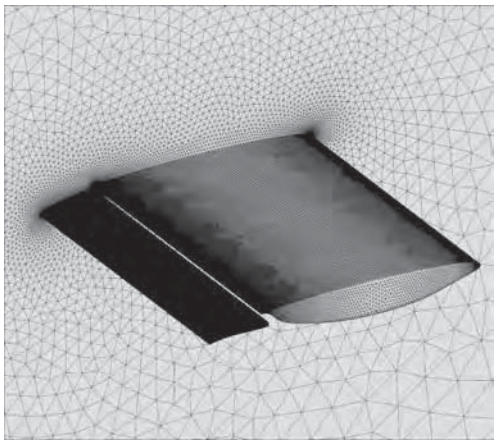


(a) Convergence history

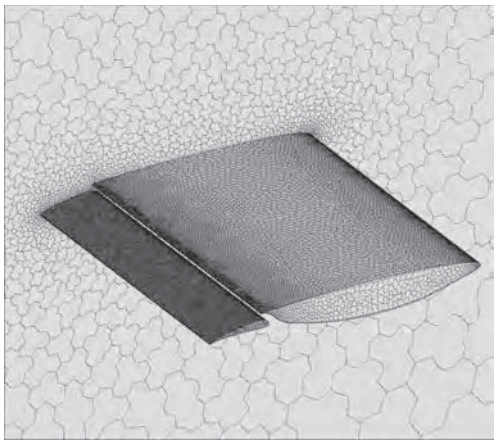


(b) Residual vs. CPU time (16 processors)

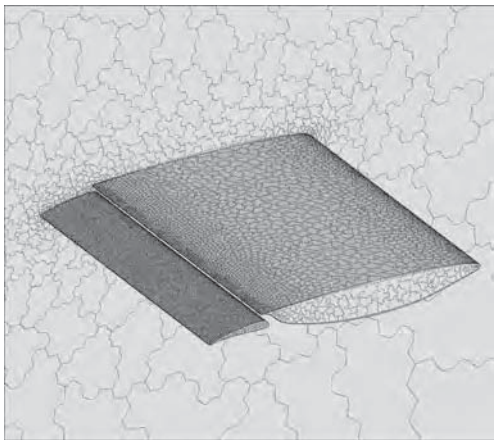
Figure 5. Convergence histories for the wing-body inviscid cases (16 processors).



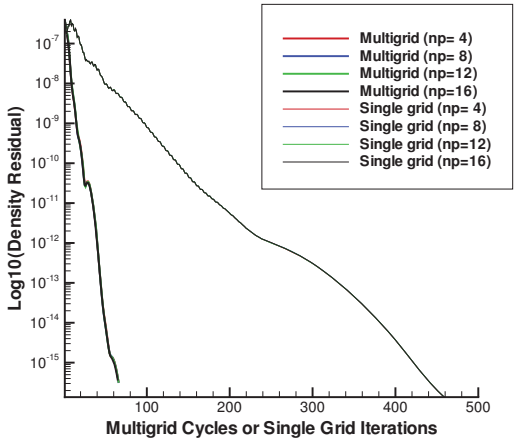
(a) Level 1: primal grid.



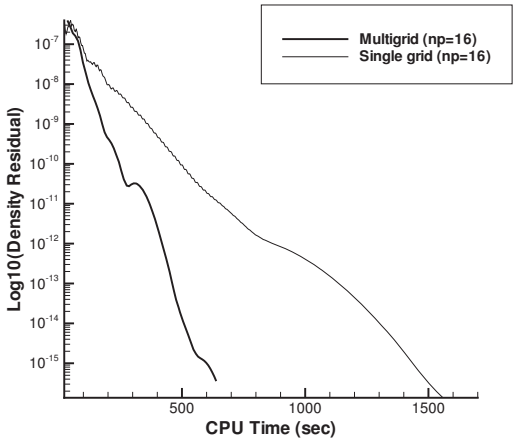
(b) Level 2: coarse grid.



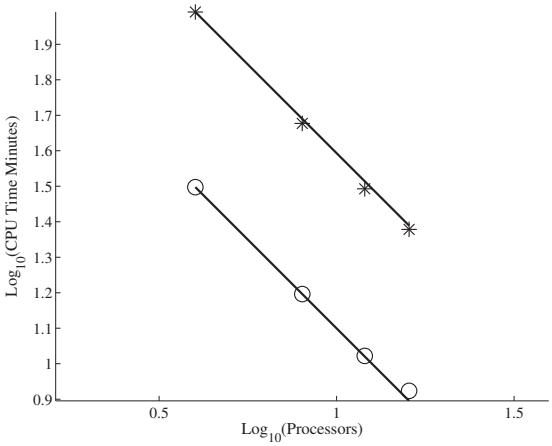
(c) Level 3: coarse grid.



(d) Convergence history (multigrid lines are overlapped; single-grid lines are overlapped.)

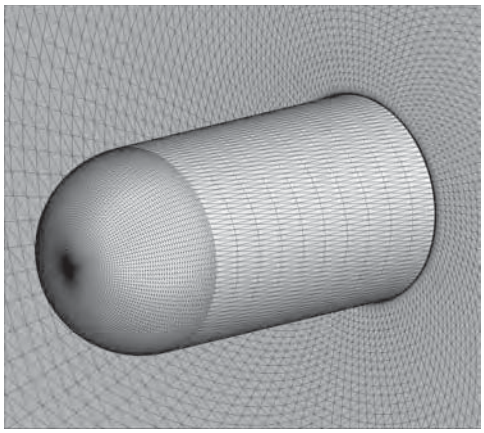


(e) Convergence history : residual vs. CPU time (16 processors)

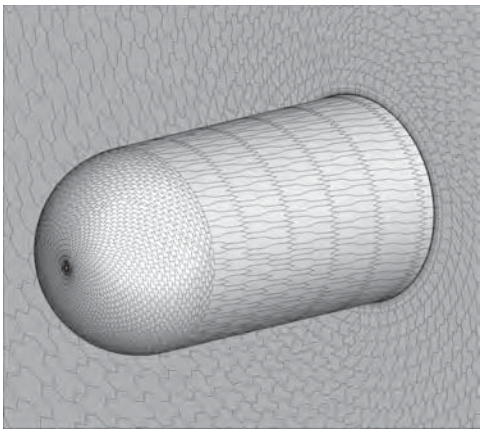


(f) Parallel Scalability (Stars for single grid; circles for multigrid; solid lines indicate the perfect scaling.)

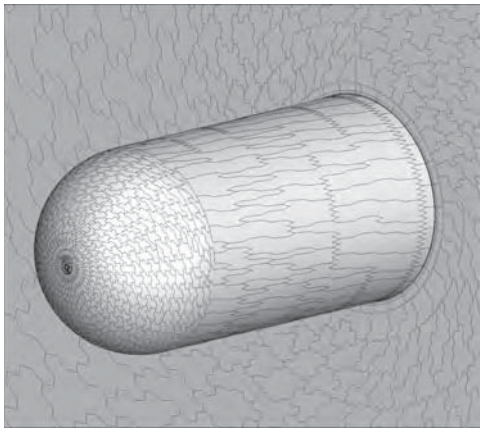
Figure 6. Grids and convergence history for the wing-flap inviscid case



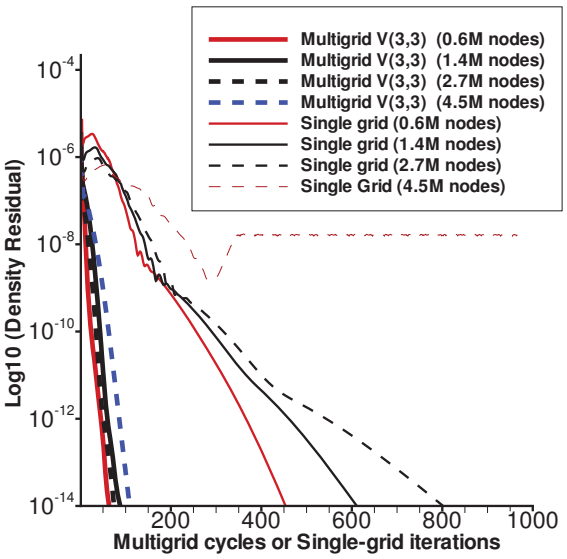
(a) Level 1: primal grid.



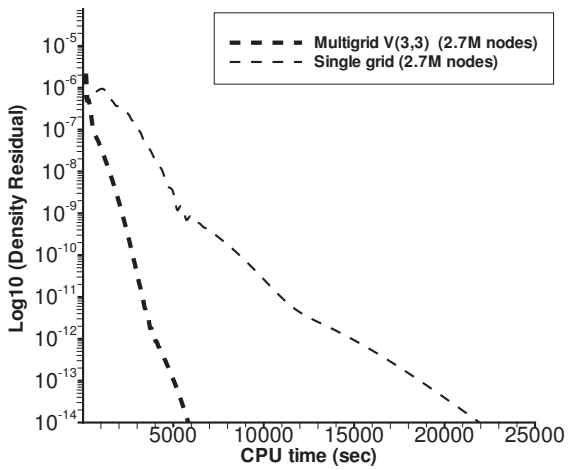
(b) Level 2: coarse grid.



(c) Level 3: coarse grid.

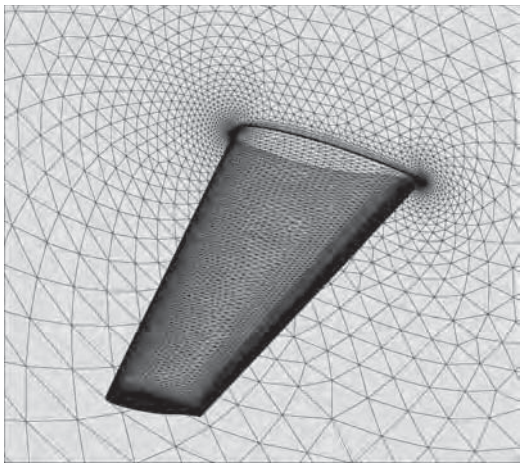


(d) Convergence history

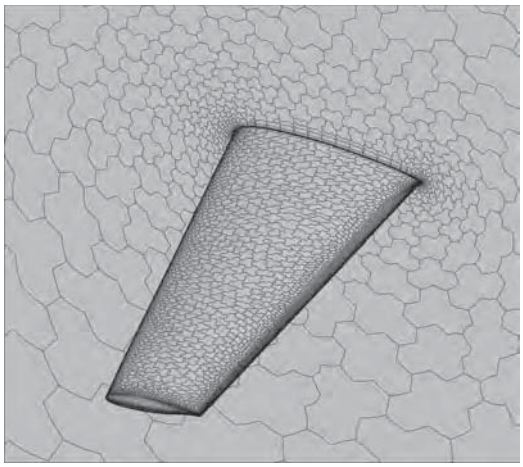


(e) Convergence history: residual vs. CPU time

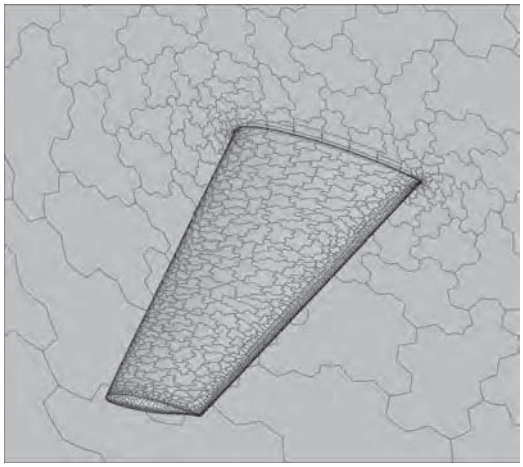
Figure 7. Grids and convergence history for the hemisphere cylinder case (Laminar; 16 processors).



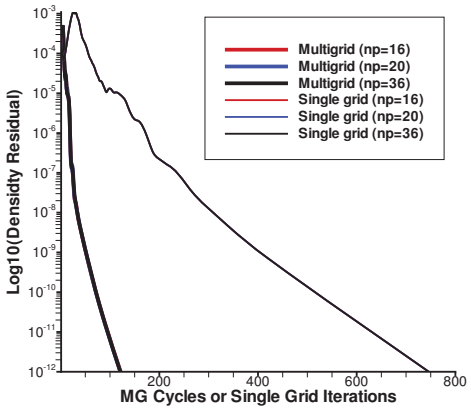
(a) Level 1: primal grid.



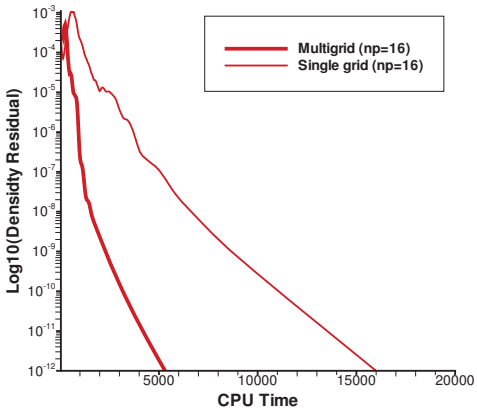
(b) Level 2: coarse grid.



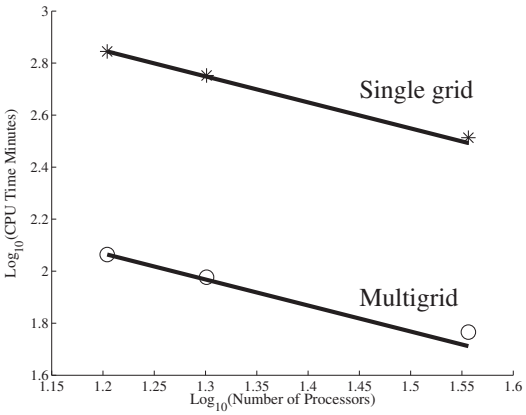
(c) Level 3: coarse grid.



(d) Convergence history: density residual vs. cycle/iteration (multigrid lines are overlapped; single-grid lines are overlapped.)

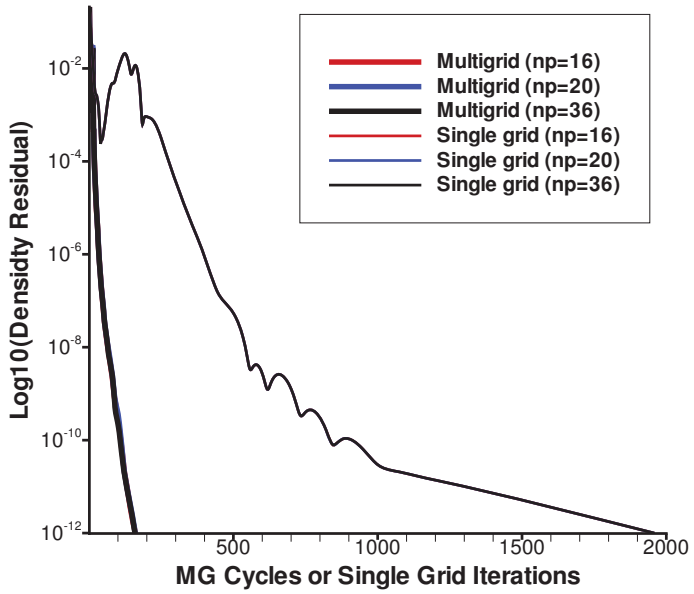


(e) Convergence history: density residual vs. CPU time (16 processors)

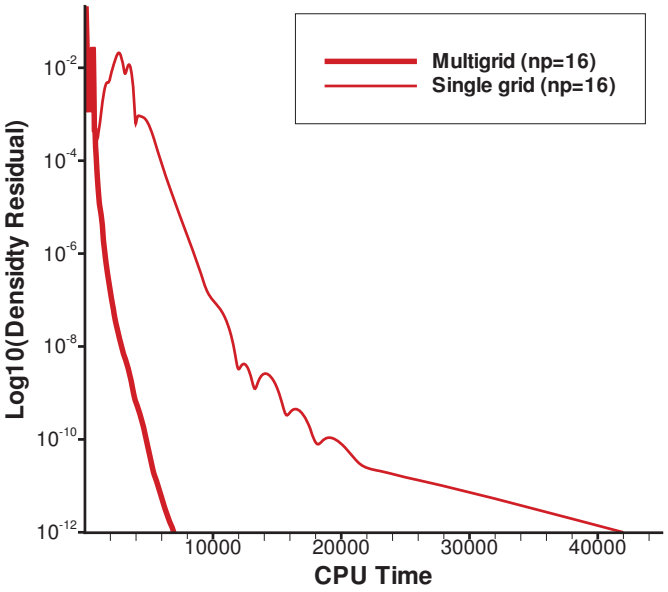


(f) Parallel Scalability (solid lines indicate the perfect scaling)

Figure 8. Grids and convergence history for the DPW-W2 case (RANS).



(a) Convergence history: turbulence residual vs. cycle/iteration (multigrid lines are overlapped; single-grid lines are overlapped.)



(b) Convergence history: turbulence residual vs. CPU time (16 processors)

Figure 9. Convergence history of the turbulence residual in the DPW-W2 case (RANS).

Accuracy of the cell-centered grid metric in the DLR TAU-Code

Axel Schwöppe and Boris Diskin

Abstract The drag prediction accuracy of the current version of the cell-centered grid metric discretization in the edge-based flow solver TAU lags behind the accuracy of the cell-vertex grid metric on highly-skewed unstructured meshes. Inaccurate convective fluxes and gradients contributing to the turbulence sources are identified as the reasons for this accuracy degradation. Alternative approaches for cell-centered discretizations are presented and shown to lead to significant accuracy and robustness improvements. Recommendations are given to improve spatial discretization schemes for the cell-centered grid metric in an edge-based finite volume code.

1 Introduction

Both cell-centered and cell-vertex discretizations are widely used for turbulent flow-simulations in aerospace applications. The relative advantages of the two approaches have been studied concerning accuracy, efficiency and robustness, but a consensus has not emerged [3, 4, 7].

The DLR RANS-Solver TAU [10] is an unstructured CFD solver based on a finite-volume discretization scheme. The geometry of a configuration is mapped by a cell-vertex grid metric and stored via an edge-based data structure. Since Release 2008.1.0 of the TAU-code, a cell-centered grid metric based on the same data structure is available as well. The drag prediction accuracy of the current cell-centered version of the TAU-Code lags behind the accuracy of the cell-vertex version for

Axel Schwöppe
Deutsches Zentrum für Luft und Raumfahrt e.V., Lilienthalplatz 7, 38108 Braunschweig, Germany, e-mail: axel.schwoeppe@dlr.de

Boris Diskin
National Institute of Aerospace (NIA), 100 Exploration Way, Hampton, VA 23681, USA, e-mail: bdiskin@nianet.org

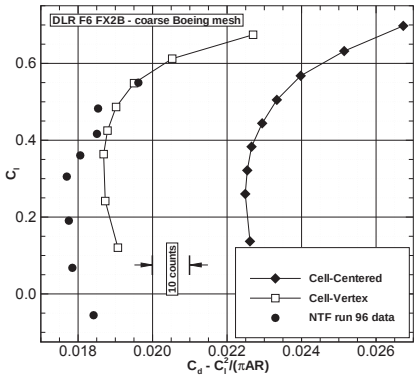


Fig. 1 Idealized drag polar of DLR F6 configuration at Mach 0.75 and Re = 3.000.000 [12] on a coarse mesh (provided by Boeing) for cell-vertex (upwind scheme, least-squares gradients) and cell-centered (upwind scheme, Green-Gauss gradients) grid metric. The original Spalart-Allmaras turbulence model is used. Circles represent wind-tunnel measurements conducted at the National Transonic Facility.

complex industrial three-dimensional (3-D) configurations. On the other side, test cases (e.g. NACA0012, flat plate) using high quality meshes, which are nearly orthogonal in relevant mesh regions, show no significant differences between grid metrics.

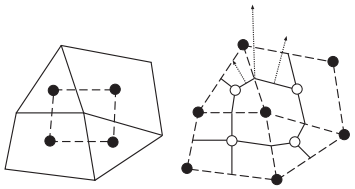
A test case from the Third AIAA Drag Prediction Workshop (DPW-III) is chosen to illustrate and explain the reasons for this accuracy degradation. The case is the DLR F6 wing-body configuration [12]. A comparison of the idealized drag polar computed on an unstructured, highly-skewed mesh is shown in Fig. 1. The mesh is the coarse mesh of the family used in the DPW-III computations for a mesh convergence study. Differences of more than 30 drag counts have been observed between the cell-vertex and the cell-centered solutions—the cell-vertex solution is in much better agreement with the wind-tunnel measurements.

This paper presents explanations for the insufficient accuracy of the cell-centered solution and offers approaches to improve this accuracy. Section 2 considers details of the spatial grid-metric discretizations relying on the edge-based data structure of the TAU-code. The gradient calculation methods used in the current TAU-code and improved approaches for the cell-centered grid metric are described in Section 3. Conclusions and recommendations for cell-centered finite volume flow solvers based on an edge-based date structure are offered in Section 4.

2 Spatial Discretization

The accuracy difference is observed in a steady case solution and, thus, has its roots in the spatial discretization. The spatial discretization used in the TAU-code is derived from the integral form of the 3-D RANS equations

$$\frac{\partial}{\partial t}\int_{\Omega} \square \, d\Omega + \oint_{\partial\Omega} (\square_c - \square_v)\, dS = \int_{\Omega} \square \, d\Omega. \qquad (1)$$



(a) Cell-Centered (b) Cell-Vertex

Fig. 2: Computational mesh for cell-vertex and cell-centered grid metric. Black circles represent locations of degrees of freedom, white circles represent vertices of the control volumes, solid lines denote faces of the control volumes, dashed lines denote edges, and arrows denote area-normal vectors.

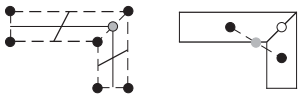
Here, t is the time, Ω is the spatial domain, $\square$ is the vector of the conservative Reynolds-averaged variables including main and turbulence variables, $\square_c$ and $\square_v$ are the respective vectors of convective and viscous fluxes, and $\square$ is the source term. The discretization of the governing e□uations follows the method of lines, which decouples the spatial and the temporal discretization [2]. The spatial domain is divided into a set of non-overlapping polyhedral control volumes, and E□ 1 is discretized for each control volume. The finite-volume discretization of E□1 at a representative control volume i can be written as

$$\frac{d\square_i}{dt} = -\frac{1}{\Omega_i}\left[\sum_{j=1}^{N}(\square_c - \square_v)_{ij}\,\mathbf{n}_{ij} - \square_i \boldsymbol{\Omega}_i\right], \tag{2}$$

where $\mathbf{n}_{ij}$ is the area-normal vector of the control volume face separating points i and j, and N is the number of face-neighbors of control volume i. The area-normal vector is the outward vector perpendicular to the face with the magnitude e□ual to the face area. The connection between point i and j is denoted as edge ij.

The set of non-overlapping polyhedral control volumes is called the computational or dual mesh. The computational mesh is dependent on the used grid metric and is based on the primary mesh, containing tetrahedrons, hexahedrons, prisms and pyramids in the context of the TAU-code. For the cell-centered grid metric, degrees of freedom are located at the centers of the primal cells. The cell center coordinates are typically defined as the averages of the cell vertex coordinates. The control volumes are the primal cells (Fig. 2(a)). For the cell-vertex grid metric, degrees of freedom are located at the vertices of the primal cells. The control volumes are constructed around the vertices by the median-dual partition: the centers of primal cells are connected with the midpoints of the surrounding faces, the area-normals can be computed as the vector sum of the area-normals of the faces ad□acent to the edge (Fig. 2(b)).

There are at least two reasons for the difference between the cell-vertex and the cell-centered solutions: (1) accuracy of the surface flux integration and (2) accuracy of the gradients contributing to the source of the turbulence e□uation. In this section, accuracy of the surface flux integration is considered□the effect of gradient approximation on the turbulent sources and solution accuracy is discussed in Section 3.



(a) Cell-Vertex (b) Cell-Centered

Fig. 3: □ossible locations of edge-midpoints (gray circles) and face-integration points (white circles) in a typical unstructured discretization, e.g. of a blunt trailing edge. Black lines represent control volume faces, dashed lines edges, black circles locations of degrees of freedom.

The surface integral of E□ 1 is approximated via the sum of fluxes over control volume faces in E□ 2. At each control volume face, the flux is reconstructed at the face-integration point and multiplied by the area-normal vector. For second-order accuracy, the reconstruction at the face-integration points should be second order accurate. In an edge-based code, the values are typically reconstructed at an edge-midpoint and used to approximate values at a face-integrations point. In a cell-vertex code, the edge-midpoints coincide with the face-integration points (Fig. 3(a)). In a cell-centered code, on highly-stretched curved grids, the locations of the corresponding edge-midpoint and face-integration point may differ significantly (Fig. 3(b)). This difference has been identified as the leading reason for inaccuracy of the order discussed in this paper.

In the TAU-code, there are two second-order schemes for the convective fluxes: a central scheme with artificial dissipation and an upwind scheme [2]. The central scheme averages flow variables and adds an artificial dissipation term to avoid odd-even decoupling.

$$\square_{ij} = \frac{1}{2}(\square_i + \square_j) + \mathbf{D}_{ij}. \tag{3}$$

Details concerning the dissipation term D_{ij} can be found e.g. in [2]. The average of the control volume values W_i and W_j is intended to provide a solution approximation at the face-integration point. In the case of the cell-centered grid metric, this average introduces an error caused by the difference between the locations of the face-integration point and edge-midpoint (Fig. 3(b)) and thus reduces the order of the scheme. This error can only be avoided if additional neighboring points are involved to get a more accurate interpolation at the face-integration point. Due to the current edge-based data structure of the TAU-code, no information about other neighboring points is available for the cell-centered metric. Thus, the central scheme is not recommended for edge-based cell-centered grid metric without altering the edge-based data structure significantly.

The upwind scheme reconstructs the fluxes at the face-integration point at the left and the right side of the face.

$$\square_{ij} = \frac{1}{2}\left(\square_L + \square_R - \left|\mathbf{A}_{ij}\right|(\square_R - \square_L)\right). \tag{4}$$

$\square_L$ and $\square_R$ are the left and right fluxes respectively, computed from the state solutions reconstructed at the corresponding side of the face, $\mathbf{A}_{ij}$ denotes the convective flux □acobian. The state solutions are reconstructed at the face-integration point

with second order using the solutions and solution gradients defined at the control-volume centers. The gradient accuracy has to be at least first order [2]. The upwind scheme, E 4, is usable for the edge-based cell-centered grid metric.

radient com utation

Two types of gradients are used in the finite-volume discretization schemes: cell gradients are used in second-order upwind schemes and in source terms for turbulence models, face gradients are used to compute viscous fluxes.

The Green-Gauss (GG) and least-s uares (LS) approaches for cell-gradient calculation are widely used. For second-order accuracy, the cell-gradient is assumed to be constant over the control volume.

Following the Green-Gauss theorem, the cell gradient is approximated as a discrete surface integral, a sum of scalar values reconstructed at the face-integration point multiplied by the area-normal face vector

$$\nabla W_i = \frac{1}{\Omega_i} \sum_{j=1}^{N} \frac{1}{2} (W_i + W_j) \cdot \mathbf{n}_{ij}. \qquad (\square)$$

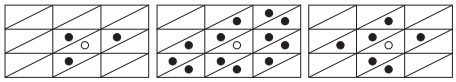
Because of the approximation properties of the cell-vertex integration scheme [,], the GG gradient is exact for a linear function only on tetrahedral or triangular meshes, although reasonable accuracy has been demonstrated in computations on mixed grids [8]. For the cell-centered metric, the GG gradient is not generally exact for a linear function accuracy is achieved only if the edge-midpoint coincides with the face-integration point [8].

The LS cell-gradient [1] is computed by solving a system of linear e uations for the gradient values. The system results from the minimization of the functional

$$\sum_{j=1}^{N} w_{ij}^2 \left(\nabla W_i \cdot \left(\square - \square_{\mathbf{i}} \right) - (W_j - W_i) \right)^2 \rightarrow \min. \qquad (6)$$

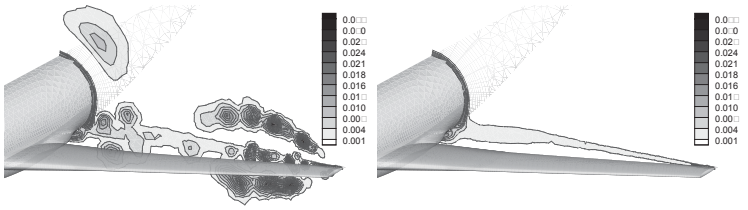
Here x_i is the coordinate vector of point i and w_{ij} is a weighting factor chosen as $w_{ij} = 1/\left| \square - \square_{\mathbf{i}} \right|$. This weighted LS method is known to improve gradient accuracy on certain high aspect ratio grids [4, 8] due to an improvement of the condition of the linear system [8]. The LS cell-gradients represent linear functions exactly for cell-vertex and cell-centered discretizations. avriplis [8] noted that this is not a sufficient criterion for accuracy certification in the context of the whole finite volume scheme. The accuracy depends on the choice of the stencils for the LS minimization.

The LS stencil is the set of points involved in the sum of E 6. A comprehensive study of inviscid finite-volume discretizations employing various LS stencils can be found in [4]. The nearest neighbor (NN) stencil includes only face-neighbors (Fig. 4(a)). The NN stencil is inexpensive, but does not necessarily provide accuracy



(a) NN (b) FA (c) SA

Fig. 4: Least-s□uare gradient stencils for cell-centered grid metric. White circle represent computation points (stencil center), black circles represent neighbors involved in the stencil.



(a) SA stencil (b) cSA stencil

Fig. □: Influence of the stencil of the least-s□uares gradient on the eddy-viscosity (plane behind the wing of the DLR-F6 configuration as in Fig. 1).

and robustness [4, 8]. The full augmentation (FA) stencil includes all neighbors that share a vertex with the given volume (Fig. 4(b)). In an edge-based code, this extension beyond the face-neighbors is straightforward. The FA stencil normally leads to robust and accurate solutions but is expensive to compute [4], in particular in 3-D cases. The smart augmentation (SA) stencil employs only a small portion of the points used on the corresponding FA stencil (Fig. 4(c)). The SA stencil expands the NN stencil by one volume point per volume vertex. In this paper for each control volume vertex, the cell center added to the SA stencil is the nearest to the stencil center of all the cells surrounding the vertex.

With this SA stencil, there still are instances, where additional points should be added to the stencil to provide accurate cell-gradients. Without sufficient cell-gradient accuracy, large errors are introduced to the turbulence e□uation through gradient sources [2], thus leading to erroneous eddy-viscosity. Non-physical vortex structures (Fig. □(a)), which have their origins at elements with inaccurate gradients, can be observed. To prevent these non-physical vortex structures, the SA stencil is expanded by adding additional points from the FA-stencil. □oints are added if their addition improves the condition number of the LS□ system. The Frobenius matrix norm is chosen to compute the condition number. The expanded stencil is denoted as conditioned smart augmentation (cSA) stencil. With the cSA stencil, the non-physical vortex structures do not appear, see Fig. □(b).

Fig. 6 shows that, with the upwind scheme using the LS□ cSA gradients, the large offset between cell-centered and cell-vertex polars has been completely removed. Note that the offset is removed even with the SA stencil□cSA stencil is re□uired to remove the non-physical vortex structures.

Face-gradients are used to evaluate the viscous fluxes □$_v$ in E□ 2. The derivatives of the velocity components and the temperature have to be known at the faces of the control volumes. The schemes for computing the face gradients strongly affect robustness of the solution process on highly-skewed meshes.

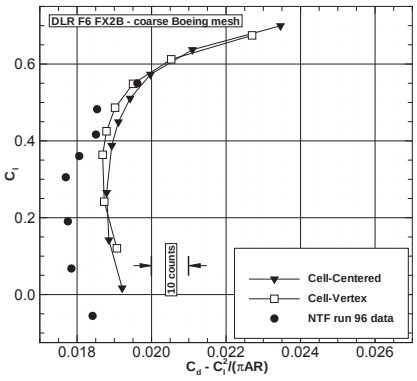


Fig. 2 Improved idealized drag polar compared to Fig. 1 for the cell-centered grid metric using second order upwind scheme, least-square gradient reconstruction based on the cSA stencil.

With an edge-based data structure, an average of the corresponding cell-gradients is typically calculated to compute the face gradients

$$\overline{\nabla W}_{ij} = \frac{1}{2}\left(\nabla W_i + \nabla W_j\right). \tag{7}$$

Hasselbacher [6] observed that such averaging leads to odd-even decoupling and introduced edge-derivative augmentation to improve robustness. It was suggested that the edge derivative can be introduced in two ways: as either edge-normal or face-tangent augmentation. The more widely used edge-normal augmentation is implemented in the TAU-code. The effects of both augmentations have been studied in [□, 11]. Face-tangent augmentation has been recommended as more robust.

The edge-normal augmentation is defined as

$$\nabla W|_{ij} = \overline{\nabla W}_{ij} - [\overline{\nabla W}_{ij} \cdot \vec{\mathbf{e}}_{ij} - \frac{W_j - W_i}{|\mathbf{e}_{ij}|}]\vec{\mathbf{e}}_{ij}, \tag{8}$$

where $\mathbf{e}_{ij}$ is the edge vector and $\vec{\mathbf{e}}_{ij}$ is the normalized edge vector. The face-tangent augmentation is defined as

$$\nabla W|_{ij} = \overline{\nabla W}_{ij} - [\overline{\nabla W}_{ij} \cdot \vec{\mathbf{e}}_{ij} - \frac{W_j - W_i}{|\mathbf{e}_{ij}|}]\frac{\vec{\mathbf{n}}_{ij}}{\vec{\mathbf{n}}_{ij} \cdot \vec{\mathbf{e}}_{ij}}, \tag{□}$$

where $\vec{\mathbf{n}}_{ij}$ is the normalized area-normal vector. Nishikawa [□] called the term in the brackets as damping term. The edge-normal augmentation leads to a non-robust scheme on highly-skewed meshes using the cell-centered grid metric. With the edge-normal augmentations, the damping-term contributions to the diffusion operator vanish when $\vec{\mathbf{n}}_{ij} \cdot \vec{\mathbf{e}}_{ij}$ approaches zero. With the face-tangent augmentation, the damping-term contributions are always large, preventing the odd-even decoupling. It has been observed that, in many cases, a converged cell-centered solution is obtained only with the face-tangent augmentation.

□ Conclusions

Inaccuracy in the cell-centered version of the edge-based TAU-code has been observed, explained, and cured. The roots of inaccuracy are twofold: (1) large deviations between the locations of the face-integration point and the edge-midpoint on non-orthogonal meshes led to accuracy deterioration in computations with a central scheme or an upwind scheme using Green-Gauss gradients for convective fluxes. (2) inaccurate gradient computations led to erroneous turbulence sources and non-physical eddy viscosity. To cure these inaccuracies, an upwind scheme using the least-s□uare gradients computed with a compact cSA stencil has been applied. Additionally, the robustness of computations has been dramatically improved by introduction of face-tangent augmentation for face-gradients used in viscous fluxes.

References

1. Barth, T., □: A 3-D Upwind Euler Solver for Unstructured □eshes. AIAA □aper □1-1□48, (1□□1)
2. Blazek, □: Computational Fluid Dynamics: □rinciples and Applications. Elsevier, Amsterdam (2001)
3. Diskin, B., Thomas, L. T., Nielsen, E. □, Nishikawa, H., White, □ A.: Comparison of Node-Centered and Cell-Centered Unstructured Finite-Volume Discretizations: Viscous Fluxes. AIAA □ournal(2010) 48(7) pp. 1326□1338. (Also AIAA 2010-□□7)
4. Diskin, B., Thomas, L. T.: Comparison of Cell-Centered and Node-Centered Unstructured Finite-Volume Discretizations: Inviscid Fluxes. AIAA 2010-107□, 48th AIAA Aerospace Sciences □eeting, □rlando, FL, □anuary (2010)
5. Diskin, B., Thomas, L. T.: Accuracy Analysis of □ixed Element Finite Volume Discretization Schemes. NIA Report 2007-08, National Institute of Aerospace, Hampton, VA 23681, USA, August (2007).
6. Haselbacher, A., Blazek, □: □n the Accurate and Efficient Discretisation of the Navier-Stokes E□uations on □ixed Grids. AIAA □aper □8-0612, (1□□8)
7. □avriplis, D.□: Aerdodynamic drag prediction using unstructured mesh solvers. CFD-Based Drag □rediction and Reduction, eds. H. Deconoinck, □. Sermus and C. van Dam, V□I Lecture Series 2003-02, von □arman Institute for Fluid Dynamics, Rhode St-Genese, Belgium, □arch (2003)
8. □avriplis, D.□: Revisiting the Least-s□uares □rocedure for Gradient Reconstruction on Unstructured □eshes. AIAA 2003-3□86, 16th AIAA CFD conference, □rlando, FL, □une (2003)
9. Nishikawa, H.: Beyond Interface Gradient: A General □rinciple for Constructing Diffusion Schemes. AIAA □aper 2010-□0□3, 40th Fluid Dynamic Conference and Exhibit, Chicago, Illinois, □une (2010)
10. Schwamborn, D., Gerhold, T., Heinrich, R.: The DLR TAU-Code: Recent Applocations in Research and Industry. ECC□□AS CFD 2006 Conference, Netherlands (2006)
11. Thomas, L. T., Diskin, B., Nishikawa, H.: A Critical Study of Agglomeration □ultigrid for Diffusion oh Highly Stretched Grids. Computers and Fluids(2010), published electronically, doi: 10.1016□□compfluid.2010.0□023
12. Vassberg, □, Broderson, □., Wahls, R., Zickuhr, T., □avriplis, D., Tinoco, E., □ani, □., Levy, D., Eisfeld, B., □urayama, □., □orrison, □: Comparison of NTF experimental data WITH CFD predictions from Third AIAA Drag □rediction Workshop, AIAA □aper 2008-6□18, □anuary (2008)

Comparison of node-centered and cell-centered unstructured finite-volume discretizations: inviscid fluxes

Boris Diskin* James L. Thomas†

Cell-centered and node-centered approaches have been compared for unstructured finite-volume discretization of inviscid fluxes. The grids range from regular grids to irregular grids, including mixed-element grids and grids with random perturbations of nodes. Accuracy, complexity, and convergence rates of defect-correction iterations are studied for eight nominally second-order accurate schemes: two node-centered schemes with weighted and unweighted least-squares (LSQ) methods for gradient reconstruction and six cell-centered schemes – two node-averaging with and without clipping and four schemes that employ different stencils for LSQ gradient reconstruction. The cell-centered nearest-neighbor (CC-NN) scheme has the lowest complexity; a version of the scheme that involves smart augmentation of the LSQ stencil (CC-SA) has only marginal complexity increase. All other schemes have larger complexity; complexity of node-centered (NC) schemes are somewhat lower than complexity of cell-centered node-averaging (CC-NA) and full-augmentation (CC-FA) schemes.

On highly anisotropic grids typical of those encountered in grid adaptation, discretization errors of five of the six cell-centered schemes converge with second order on all tested grids; the CC-NA scheme with clipping degrades solution accuracy to first order. The NC schemes converge with second order on regular and/or triangular grids and with first order on perturbed quadrilaterals and mixed-element grids. All schemes may produce large relative errors in gradient reconstruction on grids with perturbed nodes. Defect-correction iterations for schemes employing weighted least-square gradient reconstruction diverge on perturbed stretched grids. Overall, the CC-NN and CC-SA schemes offer the best options of the lowest complexity and second-order discretization errors.

On anisotropic grids over a curved body typical of turbulent flow simulations, the discretization errors converge with second order and are small for the CC-NN, CC-SA, and CC-FA schemes on all grids and for NC schemes on triangular grids; the discretization errors of the CC-NA scheme without clipping do not converge on irregular grids. Accurate gradient reconstruction can be achieved by introducing a local approximate mapping; without approximate mapping, only the NC scheme with weighted LSQ method provides accurate gradients. Defect correction iterations for the CC-NA scheme without clipping diverge; for the NC scheme with weighted LSQ method, the iterations either diverge or converge very slowly. The best option in curved geometries is the CC-SA scheme that offers low complexity, second-order discretization errors, and fast convergence.

I. Introduction

Both node-centered and cell-centered finite-volume discretization schemes are widely used for complex three-dimensional turbulent simulations in aerospace applications. The relative advantages of the two approaches have been extensively studied in the search for methods that are accurate, efficient, and robust over the broadest possible range of grid and solution parameters. The topic was discussed in a panel session at the 2007 AIAA Computational Fluid Dynamics conference, but a consensus did not emerge. One of the difficulties in assessing the two approaches is that comparative calculations were not completed in a controlled environment, i.e., computations were made with different codes and different degrees of freedom and the exact solutions were not known.

In this paper, we provide a controlled environment for comparing a subset of the discretization elements needed in turbulent simulations, namely that of the inviscid discretization. In particular, we consider a constant-coefficient

*National Institute of Aerospace (NIA), 100 Exploration Way, Hampton, VA 23681, USA, Member AIAA bdiskin@nianet.org. Also Department of Mechanical and Aerospace Engineering, University of Virginia, Charlottesville, VA 22904, USA. Supported by NASA Fundamental Aeronautics Program, Supersonics Project, NRA Contract NNL07AA23C (PI: Prof. N. K. Yamaleev)

†Computational AeroSciences Branch, NASA Langley Research Center, Mail Stop 128, Fellow AIAA, James.L.Thomas@nasa.gov.

convection equation as a model for inviscid fluxes. This paper is second in a series of papers on comparison of cell-centered and node-centered finite-volume discretizations. It follows Ref. [1], which considered viscous fluxes. The ultimate objective of the effort is to construct a uniformly second-order accurate and efficient unstructured-grid solver for the Reynolds-Averaged Navier-Stokes equations.

In this work, we use the method of manufactured solution so that the exact solution is known and conduct computational studies of accuracy, complexity, and efficiency on two-dimensional grids ranging from structured (regular) grids to irregular grids composed of arbitrary mixtures of triangles and quadrilaterals. Highly irregular grids are deliberately constructed through random perturbations of structured grids to bring out the worst possible behavior of the solution. Two classes of tests are considered. The first class of tests involves smooth manufactured solutions on both isotropic and highly anisotropic grids with discontinuous metrics, typical of those encountered in grid adaptation. The second class of tests concerns solutions and grids varying strongly anisotropically over a curved body, typical of those encountered in high-Reynolds number turbulent flow simulations.

There are eight main schemes considered — two representative node-centered schemes with weighted and unweighted least-square methods for gradient reconstruction and six cell-centered schemes. The cell-centered schemes include node-averaging schemes with and without clipping and four least-square gradient reconstruction schemes that are named according to the stencil used for the least-square fit: a nearest-neighbor scheme uses only face-neighboring cells; a smart-augmentation scheme minimally augments the nearest-neighbor stencil; two full augmentation schemes with and without weighting use larger stencils that include all node-sharing cells. Each of the schemes considered is nominally second-order accurate.

For the second class of tests, the approximately mapped least-square approach introduced in Ref. [1] is used to improve gradient reconstruction accuracy on curved high-aspect-ratio grids. The mapping uses the distance function commonly available in practical codes and can be used with any scheme.

The properties to be compared in this study are computational complexity (operation count) and discretization accuracy at equivalent numbers of degrees of freedom as well as convergence rates of defect-correction iterations with a first-order driver. The effect of clipping is studied for the node-averaging schemes.

The material in this paper is presented in the following order. Section II introduces the computational grids used in the current study. A brief explanation of finite-volume discretizations in Section III is followed by the estimates of discretization complexity for two- and three-dimensional grids given in Section IV. Section V outlines the analysis methods used in this study. A brief introduction of the model equation in Section VI precedes results provided in Section VII on accuracy of finite-volume solutions and gradients and on convergence rates of defect-correction iterations observed on isotropic irregular grids. The effect of clipping on accuracy of node-averaging schemes is also studied in this section. Section VIII compares the finite-volume discretizations on stretched highly anisotropic grids in rectangular geometries. Section IX provides comparisons for irregular high-aspect-ratio grids in curved geometries. Conclusions and recommendations are offered in Section X.

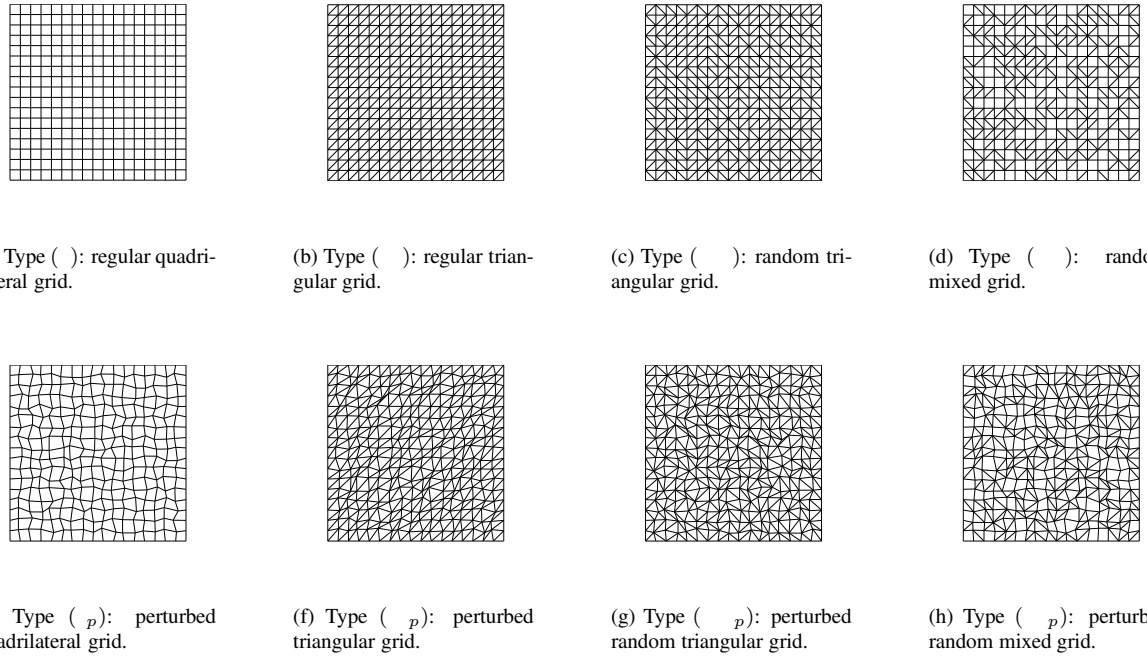
II. Grids

This paper studies finite-volume discretization (FVD) schemes for inviscid fluxes on grids that are loosely defined as irregular. A grid is classified as *regular* if it can be derived by a smooth mapping from a grid with (1) a periodic node connectivity pattern (i.e., the number of edges per node changes periodically) and (2) a periodic cell distribution (i.e., the grid is composed of periodically repeated combinations of cells). Regular grids include, but are not limited to, grids derived from Cartesian ones – triangular grids obtained by diagonal splitting with a periodic pattern, smoothly stretched grids, skewed grids, smooth curvilinear grids, etc. Grids that are not regular are called *irregular* grids. We are especially interested in *unstructured* grids, e.g., grids with the number of edges changing from node to node with no pattern.

The regular and irregular grids considered in this paper are derived from an underlying (possibly mapped) Cartesian grid with mesh sizes h_x and h_y and the aspect ratio $\mathcal{A} = h_x \ h_y$; both mesh sizes of the underlying grid are assumed to be small, $h_y \quad 1 \ h_x \quad 1$. Irregularities are introduced *locally* and do not affect grid topology and metrics outside of a few neighboring cells. A local grid perturbation is called *random* if it is independent of local perturbations introduced beyond some immediate neighborhood. For computational grids generated for the reported studies, local and random grid irregularities are introduced in two ways: (1) the quadrilateral cells of the underlying grid are randomly split (or not split) into triangles; (2) the grid nodes are perturbed from their original positions by random shifts, where the shifts are fractions of a local mesh size.

Four basic grid types are considered: (I) *regular quadrilateral* (i.e., mapped Cartesian) grids; (II) *regular tri-*

angular grids derived from the regular quadrilateral grids by the same diagonal splitting of each quadrilateral; (III) *random triangular grids*, in which regular quadrilateral are split by randomly chosen diagonals, each diagonal orientation occurring with probability of half; $(I\ \ )$ *random mixed-element grids*, in which regular quadrilateral are randomly split or not split by diagonals; the splitting probability is half; in case of splitting, each diagonal orientation is chosen with probability of half. Nodes of any basic-type grid can be perturbed from their initial positions by random shifts, thus leading to four additional *perturbed* grid types which are designated by subscript as $(I\ )$-$(I\ \ )$. Grids of types (III) $(I\ \ )$ and $(III\ )$ $(I\ \ )$ are irregular (and unstructured) because there is no periodic connectivity pattern. All perturbed grids are irregular because there is no periodic cell distribution. The representative grids are shown in Figure 1.



(a) Type ( ): regular quadrilateral grid.

(b) Type ( ): regular triangular grid.

(c) Type ( ): random triangular grid.

(d) Type ( ): random mixed grid.

(e) Type ($_p$): perturbed quadrilateral grid.

(f) Type ($_p$): perturbed triangular grid.

(g) Type ($_p$): perturbed random triangular grid.

(h) Type ($_p$): perturbed random mixed grid.

Figure 1. Typical regular and irregular grids.

Our main interest is the accuracy of FVD schemes on general irregular (mostly unstructured) grids with a minimum set of constraints. In particular, we do not require any grid smoothness, neither on individual grids nor in the limit of grid refinement. The only major requirement for a sequence of refined grids is to satisfy the *consistent refinement property*. The property requires the maximum distance across the grid cells to decrease consistently with increase of the total number of grid points, . In particular, the maximum distance should tend to zero as in 2D computations. For 3D unstructured grids, the consistent refinement property has been studied elsewhere.[2] On 2D grids, the *effective mesh size*, h_e, is computed as the norm of the square root of the control volumes.

The locations of discrete solutions are called *data points*. For consistency with the 3D terminology, the 2D cell boundaries are called faces, and the term "edge" refers to a line, possibly virtual, connecting the neighboring data points. Each face is characterized by the *directed-area vector*, which is directed outwardly normal to the face with the amplitude equal to the face area.

The random node perturbation in each dimension is defined as $- h$, where $\in [\ 1\ 1]$ is a random number and h is the local mesh size along the given dimension. With these perturbations, triangular cells in the rectangular geometry can approach zero volume. The random perturbations are introduced independently on all grids in grid refinement implying that grids of types $(I\ )$ $(I\ \ )$ are grids with discontinuous metrics, e.g., ratios of neighboring cell volumes and face areas are random on all grids and do not approach unity in the limit of grid refinement.

III. Finite-volume discretization schemes

The FVD schemes are derived from the integral form of a conservation law

$$\oint_{\partial} \mathbf{F} \quad \hat{\mathbf{n}} \quad s = \iint f \quad \Omega \tag{1}$$

where Ω is a control volume, $\mathbf{F}$ is the flux through the boundary $\quad \Omega$, $\hat{\mathbf{n}}$ is the outward unit normal vector, and f is a force function. The general FVD approach requires partitioning the domain into a set of non-overlapping control volumes and numerically implementing equation (1) over each control volume.

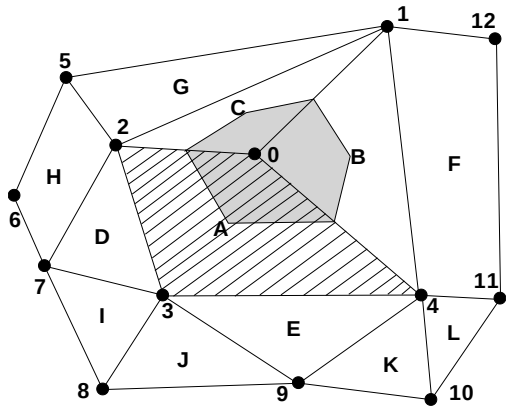


Figure 2. Control-volume partitioning for finite-volume discretizations. Numbers $0 \quad 12$ **and letters denote grid nodes and primal cell centers, respectively. The control volume for a node-centered discretization around the grid node** 0 **is shaded. The control volume for a cell-centered discretization around the cell center is hashed.**

Cell-centered (CC) discretizations assume solutions are defined at the centers of the primal grid cells with the primal cells serving as the control volumes. The cell center coordinates are typically defined as the averages of the coordinates of the cell's vertexes. Note that for mixed-element grids cell centers are not necessarily centroids. *Node-centered (NC) discretizations* assume solutions are defined at the primal mesh nodes. For NC schemes, control volumes are constructed around the mesh nodes by the median-dual partition: the centers of primal cells are connected with the midpoints of the surrounding faces. These non-overlapping control volumes cover the entire computational domain and compose a mesh that is dual to the primal mesh. Both cell-centered and node-centered control-volume partitions are illustrated in Figure 2.

The fluxes at a control-volume face are computed according to the Roe scheme,[3]

$$(\mathbf{F} \quad \hat{\mathbf{n}}) = \frac{1}{2}\left[(\mathbf{F_R} \quad \hat{\mathbf{n}}) + (\mathbf{F_L} \quad \hat{\mathbf{n}})\right] \quad \frac{1}{2}\left|\bar{\mathbf{A}}\right|(\mathbf{Q_R} \quad \mathbf{Q_L}) \tag{2}$$

where, $\mathbf{Q_L}$ and $\mathbf{Q_R}$ are the "left" and "right" solution reconstructions; $\mathbf{F_L}$ and $\mathbf{F_R}$ are the corresponding "left" and "right" numerical fluxes; $\left|\bar{\mathbf{A}}\right|$ is the Roe's approximate Riemann solver matrix. The solutions $\mathbf{Q_L}$ and $\mathbf{Q_R}$ are linearly reconstructed at the face by using solutions defined at the control volume centers and solution gradients reconstructed at each control volume. Various FVD schemes differ in the way they reconstruct gradients at the control volumes.

For cell-centered schemes, the face-based flux integration over a control-volume face is approximated as the inner product of $\mathbf{F}$ computed at the face center and the face directed area vector. The integration scheme is second-order accurate on grids of all types. For node-centered schemes, the edge-based flux integration scheme approximates the integrated flux through the two faces linked at an edge midpoint by multiplying $\mathbf{F}$ computed at the edge midpoint with the combined-directed-area vector, $\mathbf{n} = \mathbf{n_L} + \mathbf{n_R}$, where $\mathbf{n_L}$ and $\mathbf{n_R}$ are directed-area vectors of the left and right faces, respectively. The integration scheme is computationally efficient and second-order accurate on regular and triangular grids of types (I) (II) (III) $(II\ )$, and $(III\ )$; the integration accuracy degenerates to first order on mixed-element and perturbed quadrilateral grids of types $(I\ \ )$ $(I\ \ )$, and $(I\ )$.[2 5]

The forcing term integration over the control volume is approximated as the value at the control-volume center multiplied by the volume Ω . This approximation is second-order accurate when the control-volume center coincides with the centroid. On general irregular grids, the control-volume center is not necessarily the centroid, and the approximation becomes locally first-order accurate. However, with grid irregularities introduced locally and randomly (thus,

implying a zero-mean distribution of the deviations between control-volume centers and centroids), the integral of the forcing term over any sub-domain of size (1) is approximated with second order.

A. Cell-centered schemes

1. Node averaging schemes

In the cell-centered node-averaging (CC-NA) schemes, the solution values are first reconstructed at the nodes from the surrounding cell centers. With respect to Figure 2, the solution at the node 0 is reconstructed by averaging solutions defined at the cell centers A B and . The solution reconstruction proposed in Refs. [6, 7] and used in Ref. [8] is an averaging procedure that is based on a constrained optimization to satisfy some Laplacian properties. The scheme is second-order accurate and stable when the coefficients of the introduced pseudo-Laplacian operator are close to 1. It has been shown[9] that this averaging procedure is equivalent to an unweighted least-square linear fit.

The gradient at the cell Ω is reconstructed by the Green-Gauss formula,

$$\nabla \quad = \frac{1}{\Omega} \oint_{\partial} \quad \hat{\mathbf{n}} \; s \tag{3}$$

where Ω is the cell volume, $\hat{\mathbf{n}}$ is the outward unit normal, s is the area differential, and integration is performed over the cell boundary, Ω. For second-order accuracy, the solution at a face is computed by averaging the values at the face nodes and the integral over the face is approximated by the product of the solution and the face directed area.

On highly stretched and deformed grids, some coefficients of the pseudo-Laplacian may become negative or larger than 2, which has a detrimental effect on stability and robustness.[10 11] Holmes and Connell[6] proposed to enforce stability by clipping the coefficients between 0 and 2. The CC-NA schemes with clipping (CC-NA-CLIP) represent a current standard in practical computational fluid dynamics for applications involving cell-centered finite volume formulations.[12] As shown further in the paper, clipping seriously degrades accuracy of the solutions and gradients.

2. Least-square schemes

An alternative approach relies on a least-square method for gradient reconstruction, in which the linear approximation obtained at a control volume is required to coincide with the solution value at the control volume center. In this paper, both weighted and unweighted least-square methods are considered. The weighted method is designated as WLSQ herein and the unweighted method is used as default without designation. In the WLSQ method, the contributions to the minimized functional are weighted with weights inversely proportional to the distance from the control-volume center. In the unweighted method, all contributions are equally weighted.

The stencils used in the gradient fits are discussed with respect to Figure 2. Three types of stencils are considered — nearest neighbor (NN), full augmentation (FA), and smart augmentation (SA) stencils. The NN stencil involves only centers of face-neighbor cells; the FA stencil includes all the cells that share a vertex with the given cell, i.e., all the cells involved in CC-NA gradient reconstruction; the SA stencil is an adaptive stencil that provides a minimally necessary extension of the NN stencil to improve convergence rates of the defect-correction iterations (DCI) with the first-order cell-centered FVD scheme as the driver. For cell-center A, the NN stencil includes neighbors B D and ; the FA stencil includes additionally neighbors F G H I J K, and ; the SA applies an *augmentation test* to the NN stencil and expands it only if necessary and by choosing only appropriate cells from the augmentation pool provided by the FA method.

Initially, the CC-SA scheme is identical to the CC-NN scheme. In stencil augmentation at each cell, the augmentation test computes the quantity $_c = 1 \quad _{SA} \quad$, where $_{SA}$ and are the respective main-diagonal coefficients of full linearizations of the current CC-SA and the first-order driver schemes for a constant-coefficient convection operator. The test is applied for a preselected number of representative convection directions indexed by $_c$. In the algorithm implemented for this paper, the current CC-SA scheme is considered sufficiently augmented if the augmentation indicator

$$AI = \max_{c} \quad _c < \tag{4}$$

where $= 0\ 4$ is a user-defined tolerance. Smaller values of imply larger CC-SA stencils. If augmentation is required, only one cell from the augmentation pool is added to the stencil. The cells from the pool are tested one by

one until a cell that brings AI below the -threshold is found. If no such single cell has been found, the cell that makes the best improvements in AI is added to the stencil, and the augmentation procedure repeats. Note that it is possible that at the end, the user-defined tolerance has not been achieved. Even in these instances, the smart augmentation adds only cells that reduce AI, thus, providing a much smaller stencil than CC-FA stencil even in the worst-case scenario. Note, also, that the results of smart augmentation may depend on the order in which cells have been augmented. In the current paper, a sequential smart augmentation order has been used, while a fully parallel version which is independent on the augmentation order has also been developed and implemented.

B. Node-centered schemes

For the node-centered computations, the current standard employs a least-square gradient reconstruction. The typical stencil at a control volume involves all nodes linked by an edge. For example, with reference to Figure 2, the least-square fit for the shaded control volume centered at node 0 includes nodes 1 2 and 4. Both weighted and unweighted least-square methods are evaluated.

IV. Complexity

A. Flux integration complexity

In this section, the complexity associated with flux integration in 3D cell-centered or node-centered FVD schemes is estimated. The complexity is measured as the number of flux-reconstruction instances required for one residual evaluation. Flux reconstructions are the main contributers to the operation counts associated with flux integration; other aspects of the discretization, such as determining the solution values or solution-gradient values require additional considerations. Three types of primal meshes are considered: (1) fully-tetrahedral, (2) fully-prismatic, (3) fully-hexahedral.

An underlying Cartesian grid is considered and split into the various elements. The splitting into tetrahedra assumes each hexahedral defined by the grid is split into 5 tetrahedra with one of the tetrahedra being completely interior to the hexahedral (i.e., its faces are not aligned with any of the hexahedral faces – see Figure 3). Note that there are other partition strategies that lead to different number of tetrahedra per hexahedral; for example, dividing the hexahedral into two triangular prisms with subsequent division of each of the prisms into 3 tetrahedra leads to 6 tetrahedra per hexahedral. In this section we do not consider other possible partitions.

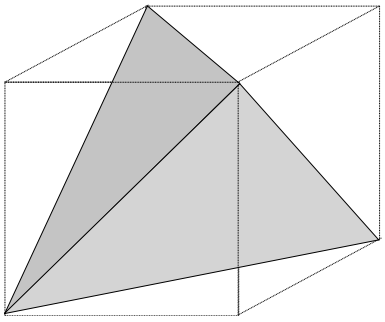


Figure 3. Splitting hexahedral into 5 tetrahedra.

Table 1 shows complexity estimates for two node-centered and one cell-centered 3D FVD schemes. Only interior discretizations are estimated; boundary effects are neglected. Both node-centered discretizations assume a median-dual partition of the domain. In such a partition, the constituent dual control volumes are bounded by generally non-planar dual faces formed by connecting 3 types of points: (1) edge midpoints, (2) element-face centroids, and (3) element centroids. FVD schemes with edge-based flux integration, such as NC schemes used in the current study, approximate integration over all of the constituent dual faces surrounding an edge midpoint by evaluating the flux at the edge midpoint; the directed area is taken as the combined directed area. FVD schemes with face-based flux integration reconstruct fluxes at each of the constituent dual faces separately and use local directed areas. For the present estimation, we assume that each flux-reconstruction instance requires the same operation count, in particular,

the approximate Riemann solver is applied at each reconstruction point. In fact, significant savings can be achieved, if the dissipation matrix is computed once for all control surfaces surrounding an edge. The first node-centered scheme is a linear 3D FVD scheme with edge-based flux integration; the second node-centered scheme is a linear 3D FVD scheme with face-based flux integration. The cell-centered formulation uses a face-based flux integration scheme with one flux reconstruction per control face.

Two estimates of complexity are given. The first estimate assumes that any constituent quadrilateral face in the control surface is broken into two triangular faces. The second estimate (in parentheses) assumes any constituent quadrilateral face is approximated as planar. The former is required to ensure a precise (water-tight) definition of the control surface and can serve as a measure of the complexity in integration of the physical flux terms. The latter can serve as an estimate of the complexity associated with numerical dissipation terms, in which details of the control-surface can be neglected.

Elements	Cell-centered face-based flux integration	Node-centered edge-based flux integration	Node-centered face-based flux integration
Tetrahedral	4 (4)	12	120 (60)
Prismatic	8 (5)	8	72 (36)
Hexahedral	12 (6)	6	48 (24)

Table 1. Number of flux-reconstruction instances per equation for 3D FVD discretizations.

The complexities of cell-centered and node-centered FVD schemes with edge-based flux integration are reasonably close. Unfortunately, as shown in this paper and also previously,[2 5] the accuracy of the edge-reconstruction FVD scheme degenerates to first order on perturbed quadrilateral and general mixed-element grids. To maintain the second-order accuracy on general grids, one can employ the node-centered scheme with face-based flux integration, but the integration complexity of this formulation substantially exceeds the complexity of the cell-centered FVD scheme. These results are in agreement with the observations made by Delanaye and Liu[13] leading to the selection of a cell-centered discretization.

B. Size of inviscid stencil

Another important measure of complexity of an FVD scheme is the size of the full-linearization stencil. The size of the 2D and 3D full-linearization stencil is examined for the inviscid cell-centered and node-centered FVD schemes. Cartesian meshes are split into triangular and tetrahedral elements, as in the previous section, again neglecting boundary effects. Estimates are compared to numerical calculations on an actual 3-D grid that includes boundary effects; the grid is a viscous fully-tetrahedral grid composed of 16,391 nodes.

In three dimensions, half of the grid nodes have 18 adjacent edges (32 adjacent tetrahedra) and half have 6 adjacent edges (8 adjacent tetrahedra). Each of the tetrahedra interior to an originally-hexahedral cell is defined by four nodes, each with 18 adjacent edges. Each of the four surrounding tetrahedra within an originally-hexahedral cell is defined by three nodes with 18 adjacent edges and 1 node with 6 adjacent edges.

For reference, Table 2 shows the average and maximum number of edges, $_{ed\ e}$, connecting to a grid node. The average number of connecting edges sets the least-square stencil size for the node-centered scheme as $_{ed\ e} + 1$. The number of connecting edges is also an important factor for the CC-NA schemes because it characterizes the number of elements sharing the node and therefore the number of cells used for averaging data to the grid node. Generally speaking, the number of edges is not bounded in 3D and, thus, the corresponding CC-NA stencil size is not bounded.

Dimension	n_{edge} (Average)	n_{edge} (Maximum)
2D	6	8
3D	12	18

Table 2. Edges connecting to a grid node in the split Cartesian grids.

For the inviscid discretization, the DCI with a first-order driver is generally used to converge the residual; thus, it is important to consider first-order and second-order linearizations. For the first-order cell-centered FVD scheme,

the size of the linearization stencil is simply the number of faces plus one (to account for the central node). For the first-order node-centered discretization, the size of the linearization stencil is the number of edges connecting to a node plus one. Table 3 shows 2D and 3D linearization stencil sizes. The cell-centered discretization has nearly a factor of 3 smaller stencil in 3D.

Elements	Node-centered	Cell-centered
Estimate 2D	7	4
Estimate 3D	13	5
Numerical 3D	14	5

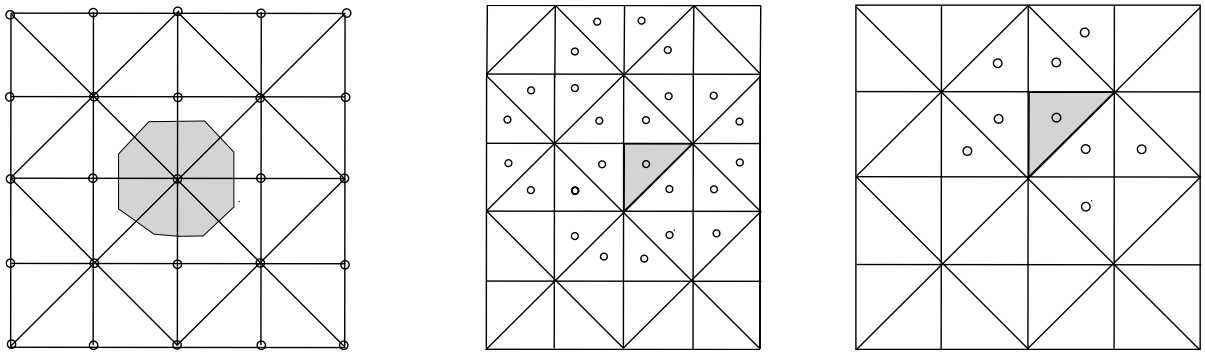
Table 3. Average size of the inviscid first-order FVD stencil on triangular/tetrahedral grids in 2D/3D.

For second-order accuracy, all schemes reconstruct gradients at the control volumes. The node-centered discretizations use a least-squares approach and require solutions at the neighbor-of-neighbor nodes and a correspondingly large linearization stencil. The cell-centered CC-NA schemes have even larger linearization stencils which include all cells contributing to solution reconstruction at any node of a face-neighboring cell. Stencils of CC-FA schemes are the same as CC-NA stencils. The CC-NN stencil also uses a least-squares approach to fitting the gradient in reconstruction, but requires a much smaller stencil which includes only neighbor-of-neighbor cells. Table 4 shows stencil sizes for 2D and 3D; in 3D, only the splitting shown in Figure 4 is considered. In three-dimensions, the NC stencil is significantly smaller than the CC-NA and CC-FA stencils. In both 2D and 3D, the CC-NN stencil is the smallest.

Elements	NC	CC-NA	CC-NN
Estimate 2D	23	25	9
Estimate 3D	75	139	15
Numerical 3D	63	118	15

Table 4. Average size of the inviscid second-order stencil for 2D/3D discretizations with triangular/tetrahedral elements.

The numbers are so striking that it is useful to show the stencils for a single shaded control volume in Figure 4 for each approach. The stencil sizes are 25, 25, and 9 for the NC, CC-NA, and CC-NN schemes, respectively. Note that the stencil size for the NC control-volume adjacent to the one shown in Figure 4 is 21; thus,the average of 23 is shown in Table 4. Also, for the 3D NC schemes, the nodes with 6 and 18 edges have stencil sizes of 57 and 93, respectively; thus, the average of 75 is shown in the table. For the CC-NA and CC-FA schemes, the cells at the corners of the original Cartesian cell have a stencil size of 149 and those fully interior to the original Cartesian cell have a stencil size of 99. Since there is one interior tetrahedron for each of the four corner tetrahedrons, the average of 139 is shown in the table.



(a) NC scheme. (b) CC-FA and CC-NA schemes. (c) CC-NN scheme.

Figure 4. Inviscid 2D stencil for shaded control volume.

V. Analysis

A. Method of manufactured solution

Accuracy of FVD schemes is analyzed for known exact or manufactured solutions. The forcing function and boundary values are found by substituting this solution into the governing equations, including boundary conditions. The discrete forcing function is defined at the data points.

1. *Discretization error*

The main accuracy measure is the *discretization error*, $_d$, which is defined as the difference between the exact discrete solution, $^{-h}$, of the discretized equations (1) and the exact continuous solution, , to the corresponding differential equations

$$_d = \quad ^{-h}; \tag{5}$$

is sampled at data points.

2. *Truncation error*

Another accuracy measure commonly used in computations is *truncation error*. Truncation error, , characterizes the local accuracy of approximating the differential equations. For finite differences, it is defined as the residual obtained after substituting the exact solution into the discretized differential equations.[1] For FVD schemes, the traditional truncation error is usually defined from the time-dependent standpoint.[15 16] In the steady-state limit, it is defined (e.g., in Ref. [17]) as the residual computed after substituting into the normalized discrete equations (1),

$$= \frac{1}{\Omega}\left[\iint f^h \ \ \Omega + \oint_{\partial} \ \mathbf{F^h} \ \ \hat{\mathbf{n}}) \ \ s \right] \tag{6}$$

where Ω is the measure of the control volume,

$$\Omega = \iint \ \Omega \tag{7}$$

$\mathbf{F^h}$ is a numerical flux evaluated at the control-volume boundary Ω, f^h is an approximation of the forcing function f on Ω, and the integrals are computed according to some quadrature formulas. Note that convergence of truncation errors is expected to show the order property only on regular grids; on irregular grids, it has been long known that the design-order discretization-error convergence can be achieved even when truncation errors exhibit a lower-order convergence or, in some cases, do not converge at all.[17 21]

3. *Accuracy of gradient reconstruction*

Yet another important accuracy measure is the accuracy of gradient approximation at a control-volume. For second-order convergence of discretization errors, the gradient is usually required to be approximated with at least first order. For each control-volume, accuracy of the gradient is evaluated by comparing the reconstructed gradient, ∇_r, with the exact gradient, $\nabla \ _{\mathrm{c}}$, computed at the control-volume center. The accuracy of gradient reconstruction is measured as the relative gradient error:

$$= \frac{}{G} \tag{8}$$

where functions and G are amplitudes of the gradient error and the exact gradient, respectively, evaluated at face centers;

$$= \ \nabla_r \ ^h \quad \nabla \ _{\mathrm{c}} \qquad \text{and} \qquad G = \ \nabla \ _{\mathrm{c}} \ ; \tag{9}$$

and ^{h} are a differentiable manufactured solution and its discrete representation (usually injection) on a given grid, respectively; is a norm of interest computed over the entire computational domain.

4. Convergence of iterative solvers

Besides accuracy, an important quality of a practical discretization is availability of an affordable solver. For FVD schemes with low complexity, such as CC-NN and CC-SA, an efficient solution method would use a full linearization in relaxation of the target FVD scheme. For FVD schemes with high complexity, such as CC-NA, CC-FA, and even NC schemes, iterations with the full linearization are not affordable; DCI schemes with linearized first-order drivers are common methods used in practical computations. In this view, stability and convergence rates of DCI are also analyzed. Let ^{h} be the current solution approximation. The DCI method is defined in the following two steps:

1. The correction ^{h} is calculated from
$$\overset{h}{d}\ ^h = \quad ^h \quad ^h\big) \tag{10}$$
where $^h \quad ^h\big)$ is the residual of the target FVD scheme and $\overset{h}{d}$ is a driver scheme.
2. The current approximation is corrected
$$^h = \ ^h + \ ^h \tag{11}$$

All considered second-order FVD schemes use the first-order upwind FVD scheme as a driver.

VI. Convection equation

The linear convection equation
$$(\mathbf{a} \ \nabla) \quad = f \tag{12}$$
is considered as a model for inviscid fluxes; $\mathbf{a}$ is a vector-function of spatial variables. The forcing function f is independent of the solution . Boundary conditions are typically defined either in a weak form as the normal flux, $(\mathbf{F} \ \hat{\mathbf{n}}) = \ (\mathbf{a} \ \hat{\mathbf{n}})$, given at the inflow boundary or as over-specified conditions, in which solutions at control volumes that include nodes edge-connected to the boundary are over-specified from the manufactured solution. In the tests reported further in this paper, the convection direction is constant, $\mathbf{a} = \ \sin \ —\big) \ \cos \ —\big)\big)$, and boundary conditions are over-specified.

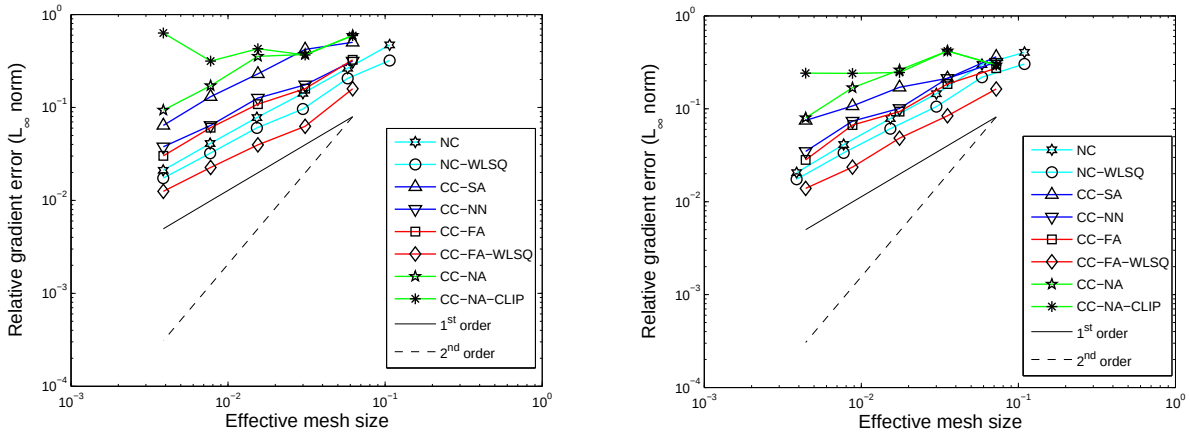
VII. Isotropic irregular grids

A. Grid refinement

All computations in this section are performed for for the manufactured solution $= \ \cos(2 \ x \quad y)$. Sequences of consistently refined grids of types $(III\ )$ and $(I \quad)$ are generated on the unit square $[0 \ 1] \quad [0 \ 1]$. Irregularities are introduced at each grid independently, so the grid metrics remain discontinuous on all the grids. The ratio of areas of neighboring faces can be as large as $3\sqrt{2}$; because a control volume can be arbitrarily small, the ratio of the neighboring volumes can be arbitrarily high. Two node-centered and six cell-centered schemes are considered: NC, NC-WLSQ, CC-SA, CC-NN, CC-FA, CC-FA-WLSQ, CC-NA and CC-NA-CLIP. On grids of type $(III\ )$, CC-SA scheme augments about 50% of the interior least-square stencils and CC-NA-CLIP clips about 10% of the interior nodes. On grids of type $(I \quad)$, CC-SA scheme augments between 25% and 30% of the interior least-square stencils and CC-NA-CLIP clips about 3% of the interior nodes. On grids of both types, about 80% of the augmented stencils increase the stencil size just by one cell, about 20% by 2 cells, and less than 1% by more than 2 cells.

B. Gradient reconstruction accuracy

For second-order discretization accuracy, the gradient reconstruction is required to be at least first-order accurate. To evaluate the gradient reconstruction accuracy, the computational gradients have been reconstructed within interior control volumes from the manufactured solution evaluated at the data points and compared with the exact gradients computed at the control-volume centers. Figure 5 shows convergence of the norms of relative gradient errors on grids of types $(III\ )$ and $(I \quad)$. Only errors computed with the CC-NA-CLIP scheme do not converge in grid refinement. Similar absence of convergence has been observed and reported previously[1] for gradients reconstructed with the clipped CC-NA scheme within control-volume faces. All other methods provide first-order gradient approximations on grids of both types.



(a) Grids of type ($_p$).

(b) Grids of type ($_p$).

Figure 5. Accuracy of gradient reconstruction for cell-centered FVD schemes on isotropic irregular grids. Manufactured solution is $= \cos(2 \quad)$.

C. Convergence of truncation and discretization error

Numerical tests evaluating convergence of truncation and discretization errors are performed for the constant-coefficient convection equation (12). Figures 6 and 7 show convergence of the norms of truncation and discretization errors, respectively.

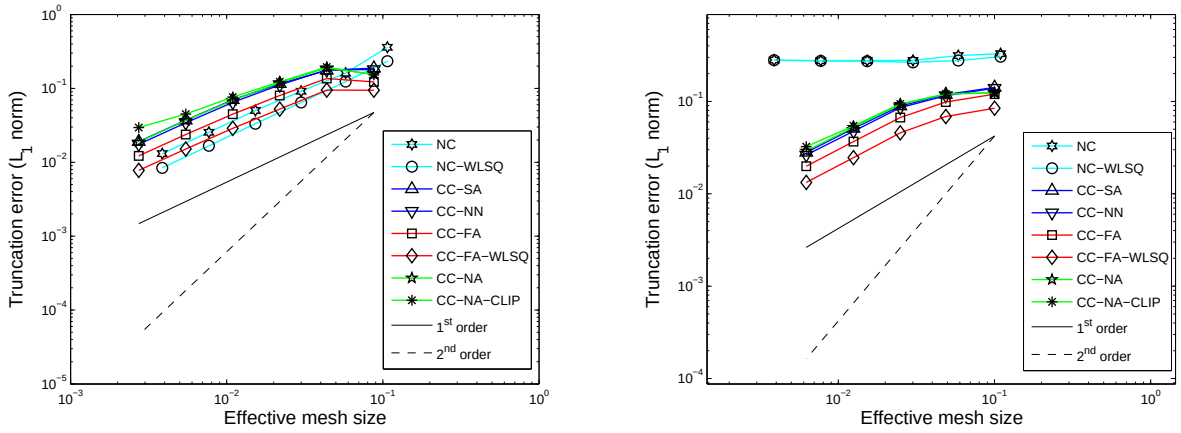
Truncation errors of all the cell-centered schemes (except the CC-NA-CLIP scheme) converge with first order on grids of both types and truncation errors of the node-centered schemes converge with first order on triangular grids of type (III); the corresponding discretization errors converge with second order. As predicted in Refs. [2,5], truncation errors of node-centered schemes do not converge on mixed-element grids; discretization errors converge with first order. The reason for this convergence degradation is the edge-based flux integration scheme, which is second-order accurate on simplex (triangular and tetrahedral) grids, but only first-order accurate on perturbed quadrilateral and general mixed-element grids. As shown in Ref. [5], with a more accurate face-based flux integration scheme, second-order accuracy is achieved with NC schemes on arbitrary grids. Although barely discernible, convergence of truncation and discretization errors of the CC-NA-CLIP scheme deteriorates on finer grids. Detailed tests performed on finer grids and reported in a subsequent section show that truncation error convergence stagnates and discretization error convergence deteriorates to first order. Also not shown, convergence of the norms of the CC-NA-CLIP scheme show signs of deterioration on coarser grids. For other schemes, convergence slopes are the same for all norms and do not change on finer grids.

All second-order discretization error plots are very close to each other indicating similar accuracy on grids with equivalent number of degrees of freedom. For reference, Figures 7(a) and 7(b) include the convergence plots of "ideal" discretization errors computed with the CC-EG scheme that uses exact gradients evaluated at each cell from the manufactured solution. These plots represent the best-possible second-order convergence, which can be achieved on given grids. Close proximity of the actual and the ideal second-order discretization errors indicates that the accuracy is nearly optimal.

D. Convergence of defect-correction iterations

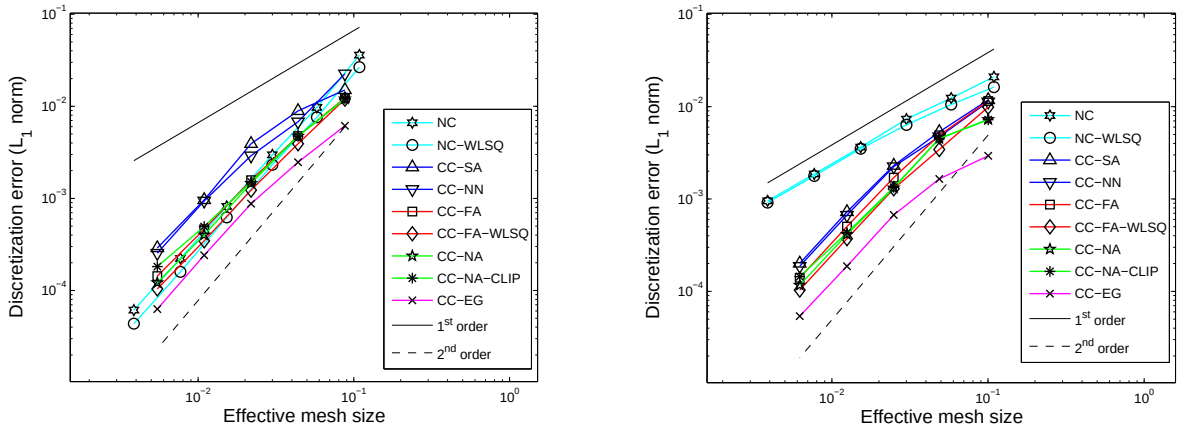
Convergence of DCI is studied for the second-order FVD schemes on isotropic grids of types (III) and $(I \quad)$ with 65 nodes. The forcing term and the boundary conditions are set to zero. The initial solution is random. Convergence rates are shown in Figure 8. As was mentioned above, the CC-SA and CC-NN schemes have small stencils and can be relaxed with full linearization of target second-order operators. However for consistency, convergence rates of DCI are shown for these schemes as well.

The DCI method for all schemes converges fast with an average convergence rate per iteration better than 0 6. The



(a) Grids of type ($_p$). (b) Grids of type ($_p$)

Figure 6. Convergence of $_1$-norms of truncation errors of FVD schemes on irregular grids. Manufactured solution is $=$ $\cos(2$).



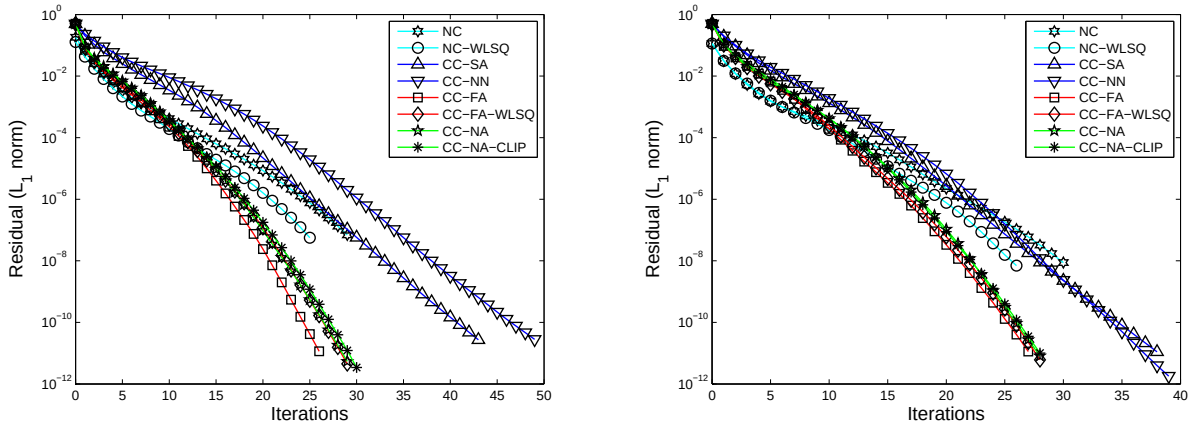
(a) Grids of type ($_p$). (b) Grids of type ($_p$)

Figure 7. Convergence of $_1$-norms of discretization errors of FVD schemes on irregular grids of of types ($_p$) and ($_p$). Manufactured solution is $=$ $\cos(2$).

convergence plots can be divided into three parts: initial convergence, transition, and asymptotic convergence. Initial convergence is typically fast for random initial solutions. The number of iterations transitions within the transition region grows slightly on finer grids. Asymptotic convergence rates for all schemes are around $0\ 5$ per iteration. Note, that on grids of type (I), all studied discretization schemes correspond to the Fromm discretization of the convection equation. A detailed study of DCI for the Fromm discretization on Cartesian grids has been reported elsewhere.[22] Note, also, that reported problems with stability of DCI for the WLSQ schemes[23] and for the CC-NA scheme without clipping[6] are not evident on these isotropic grids.

E. Effects of clipping

The tests reported in this section are performed for the CC-NA and CC-NA-CLIP schemes and demonstrate detrimental effects of clipping on convergence of gradient-reconstruction, truncation, and discretization errors in grid refinement.



(a) Grids of type ($_p$).

(b) Grids of type ($_p$)

Figure 8. Convergence of $_1$-norms of residuals in DCI for second-order FVD schemes with first-order drivers on isotropic irregular grids of types ($_p$) and ($_p$).

Considered irregular triangular grids of type $(III\ )$ are characterized by a higher percentage of clipped nodes; about 10% of the interior nodes are clipped. Figure 9(a) shows an example of a grid of type $(III\ )$ with 17 nodes; nodes where clipping occurs are circled.

Figure 9(b) shows that the gradients reconstructed by the CC-NA-CLIP scheme do not approximate the exact gradients. The CC-NA scheme provides a first-order accurate gradient reconstruction, which is sufficient for second-order discretization accuracy. Figures 9(c) and 9(d) exhibit convergence of the norms of truncation and discretization errors, respectively. The CC-NA scheme demonstrates first-order convergence of truncation errors and second-order convergence of discretization errors. Truncation errors are very similar on coarse grids, but start to diverge on finer grids. Truncations errors of the CC-NA scheme demonstrate clear first-order convergence; truncation errors of the CC-NA-CLIP scheme converge slower on finer grids and eventually stagnate. The discretization error convergence of the CC-NA-CLIP scheme exibits second order on the coarse grids, but then degrades to first order. Although not shown, the norm of discretization errors of the CC-NA-CLIP scheme shows degradation on coarser grids in grid refinement; asymptotically, norms of both node-averaging schemes converge with the same orders as the corresponding norms. Note that on grids with a small percentage of clipped nodes, convergence degradation becomes visible only on very fine grids. This may explain why such degradation has not been reported for practical computations.

VIII. Anisotropic irregular grids

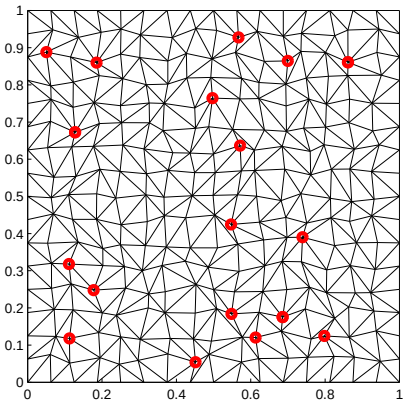
A. Grid stretching

In this section, we study FVD schemes on stretched grids generated on rectangular domains. Figure 10 shows an example grid of type $(III\ )$ with the maximal aspect ratio $\mathcal{A} = 10$. The manufactured solution is $= \sin(\ x+2\ y)$. A sequence of consistently refined stretched grids is generated on the rectangle $(x\ y) \in [0\ 1] \quad [0\ 0\ 5]$ in the following 3 steps.

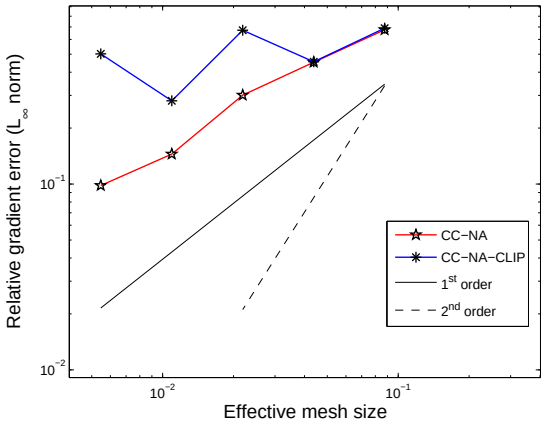
1. A background regular rectangular grid with $= (\ _x + 1) \quad (\ _y + 1)$ nodes and the horizontal mesh spacing $h_x = \frac{}{_x}$ is stretched toward the horizontal line $y = 0\ 25$. The y-coordinates of the horizontal grid lines in the top half of the domain are defined as

$$y_{\underline{N_y}} = 0\ 25; \qquad y\ = y\ + \hat{h_y} \quad \left(\frac{N_y}{}\right) \qquad = \frac{y}{2} + 2 \qquad _y \quad _y + 1 \tag{13}$$

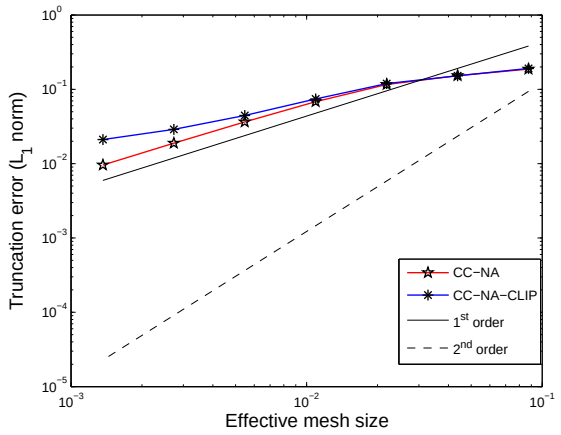
Here $\hat{h_y} = \frac{h_x}{A}$ is the minimal mesh spacing between the vertical lines; $\mathcal{A} = 10$ is a fixed maximal aspect ratio; is a stretching factor, which is found from the condition $y\ _y = 1$. The stretching in the bottom half of the



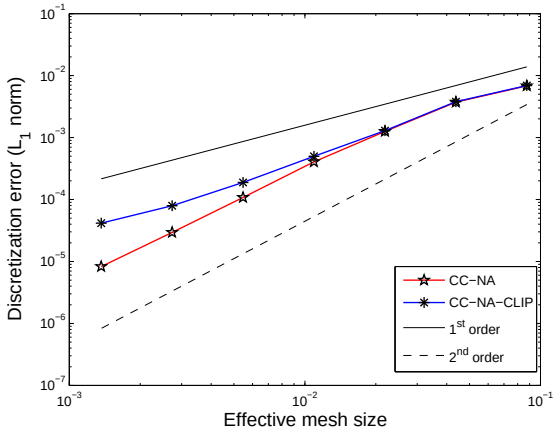
(a) Random triangular grid with 17^2 nodes. Clipped nodes are circled.



(b) Gradient errors.



(c) Truncation errors.



(d) Discretization errors.

Figure 9. Accuracy of CC-NA schemes on isotropic irregular triangular grids of type ($_p$). Manufactured solution is $=$ $\cos(2$).

domain is defined analogously.

2. Irregularities are introduced by random shifts of interior nodes in the vertical and horizontal directions. The vertical shift is defined as $\Delta y = - \min(h_y \quad h_y)$, where is a random number between 1 and 1, and h_y and h_y are vertical mesh spacings on the background stretched mesh around the grid node. The horizontal shift is introduced analogously, $\Delta x = - h_x$. With these random node perturbations, all perturbed quadrilateral cells are convex.

3. Each perturbed quadrilateral is randomly triangulated with one of the two diagonal choices; each choice occurs with a probability of one half.

B. Gradient reconstruction accuracy

A recent study[2] assessed accuracy of gradient approximation on various irregular grids with high aspect ratio $\mathcal{A} = \frac{h_x}{h_y} \gg 1$. The study indicates that for rectangular geometries and functions predominantly varying in the direction of

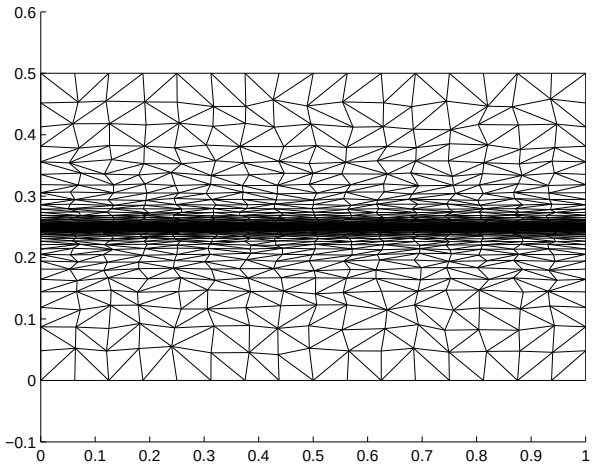
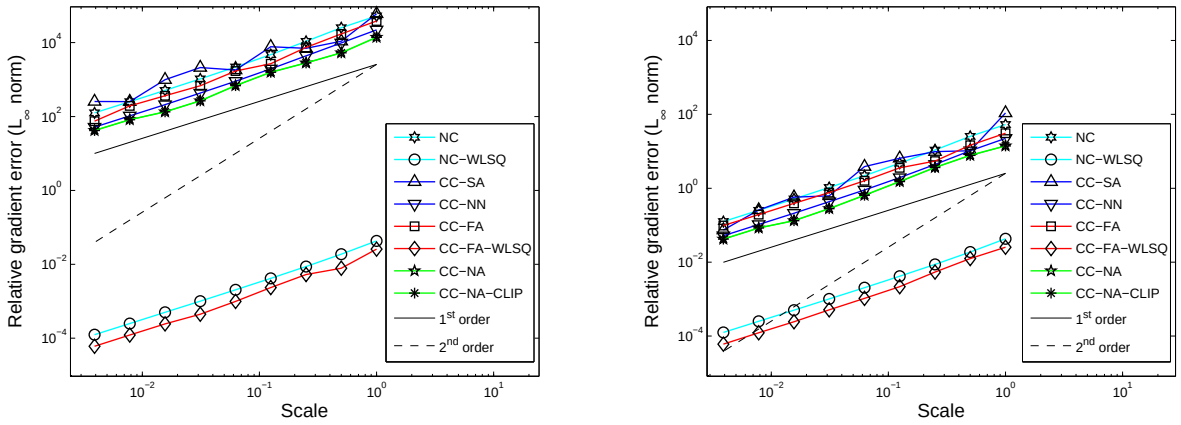


Figure 10. Random triangular stretched grid with 17 65 **nodes.**

small mesh spacing (y-direction), gradient reconstruction is accurate. For manufactured solutions significantly varying in the direction of larger mesh spacing (x-direction), the gradient reconstruction may produce extremely large $(\mathcal{A}h_x)$ relative errors affecting the accuracy of the y-directional gradient component. Figure 11 shows examples of first-order accurate gradient approximations that exhibit large relative errors on high-aspect-ratio grids of type (III).



(a) Aspect ratio $\mathcal{A} = 10^6$.

(b) Aspect ratio $\mathcal{A} = 10^3$.

Figure 11. Relative errors in approximation of gradients for the manufactured solution = sin(+ 2) **on anisotropic grids of type** ( ) **downscaled toward the focal point** ( ) = (0 3 0 5)**.**

Evaluation of gradient reconstruction accuracy is performed with the methodology of downscaling described in detail elsewhere.[2 5] The computational tests are performed on a sequence of downscaled narrow domains ($\mathcal{A}$) centered at the focal point $(x\ y) = (0\ 3\ 0\ 5)$. The scale changes as = 2 = 0 8 and the considered aspect ratios are $\mathcal{A} = 10$ and $\mathcal{A} = 10$; the latter corresponds to the highest aspect ratio observed at the central line of the stretched grid shown in Figure 10. On each domain, an independent high-aspect-ratio random grid of type (III) with 9 nodes is generated; the grid aspect ratio is fixed as $\mathcal{A}$ on all scales. The gradient reconstruction accuracy was measured at the interior control volumes. Only weighted-least-square schemes, NC-WLSQ and CC-FA-WLSQ, provide accurate gradients, the relative errors of gradient reconstructions provided by all other schemes are several

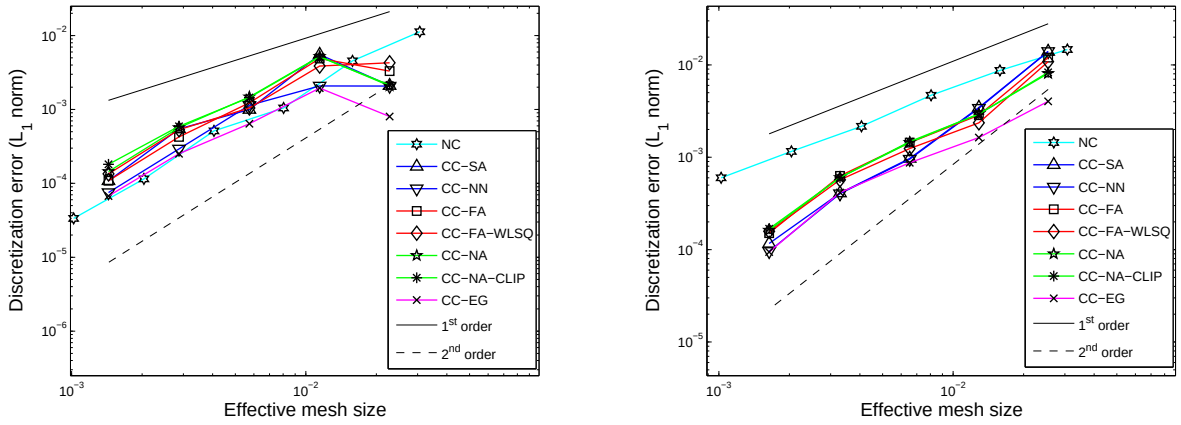
orders of magnitude larger, directly proportional to the aspect ratio $\mathcal{A}$, and converge with first order.

A summary of the results concerned with gradient accuracy on anisotropic grids is presented in Table 5. All considered gradient reconstruction methods may generate large relative errors on perturbed grids of types $(I\ )$ $(I\ \ )$. Only the NC-WLSQ scheme provides gradient reconstruction accuracy on all unperturbed grids. On perturbed grids, there are topologies, where all stencil points are almost equidistant from the stencil center, and the WLSQ method is ineffective. Such situations occur more frequently for cell-centered schemes; all cell-centered schemes may generate large gradient errors even on unperturbed mixed-element grids of type $(I\ \ )$. The CC-NN, CC-NA, and CC-FA-unweighted methods may also have large relative errors on random triangular grids of types (III); the CC-FA-WLSQ method always provides accurate gradients on these grids.

Table 5. Relative error of gradient reconstruction.

Grids	(I)	(II)	(III)	$(I\ \ )$	$(I\ )$ $(I\ \ )$
NC	(h_x)	(h_x)	$(\mathcal{A}h_x)$	$(\mathcal{A}h_x)$	$(\mathcal{A}h_x)$
NC-WLSQ	(h_x)	(h_x)	(h_x)	(h_x)	$(\mathcal{A}h_x)$
CC-SA	(h_x)	(h_x)	$(\mathcal{A}h_x)$	$(\mathcal{A}h_x)$	$(\mathcal{A}h_x)$
CC-NN	(h_x)	(h_x)	$(\mathcal{A}h_x)$	$(\mathcal{A}h_x)$	$(\mathcal{A}h_x)$
CC-FA-unweighted	(h_x)	(h_x)	$(\mathcal{A}h_x)$	$(\mathcal{A}h_x)$	$(\mathcal{A}h_x)$
CC-FA-weighted	(h_x)	(h_x)	(h_x)	$(\mathcal{A}h_x)$	$(\mathcal{A}h_x)$
CC-NA	(h_x)	(h_x)	$(\mathcal{A}h_x)$	$(\mathcal{A}h_x)$	$(\mathcal{A}h_x)$

C. Convergence of discretization errors



(a) Grids of type ($_p$). (b) Grids of type ($_p$)

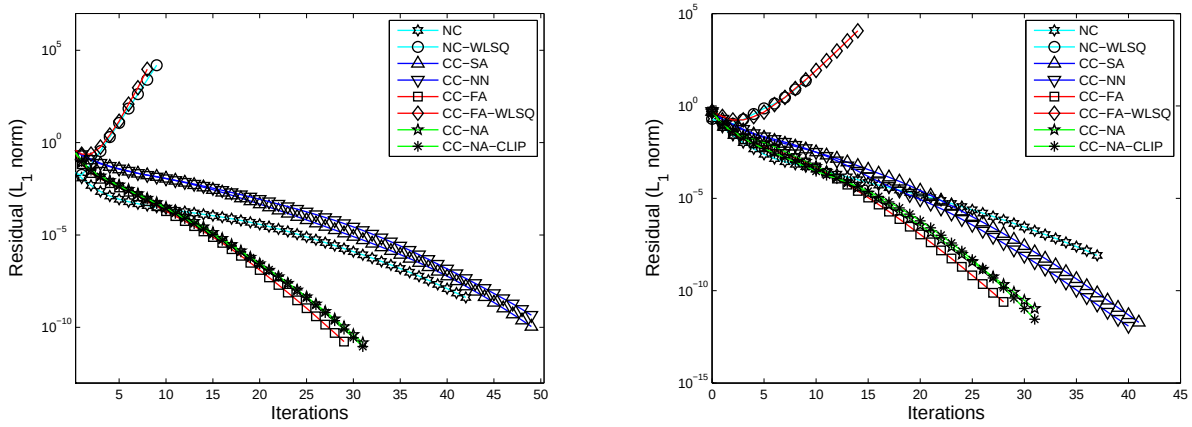
Figure 12. Convergence of discretization errors for solution $= \sin(\ \ + 2\ \ )$ **on stretched grids of types** ($_p$) **and** ($_p$)**.**

A poor gradient reconstruction accuracy, however, does not necessarily imply large discretization error. Second-order accurate solutions have been previously reported[1 25] on grids with large gradient reconstruction errors. Here, we observe similar results for cell-centered and node-centered FVD schemes for constant-coefficient convection. Convergence histories of the norms of discretization errors for the manufactured solution $= \sin(\ x + 2\ y)$ on a sequence of consistently refined stretched grids of types $(III\ )$ and $(I\ \ )$ are shown in Figure 12. On grids of type $(III\ )$, all discretization errors converge with second order. Note that, from the convergence results reported in Section VII (subsection E), discretization-error convergence order for the CC-NA-CLIP scheme is expected to deteriorate

to first order on finer grids. Discretization errors of the NC-WLSQ scheme are not shown in Figure 12 because the NC-WLSQ scheme does not converge in DCI on grids of types $(III\ )$ and $(I\ \ )$. The NC scheme converges with first order, as expected. Discretization errors of all cell-centered schemes converge with second order, close to each other and to the ideal discretization errors (CC-EG).

D. Convergence of defect-correction iterations

The DCI method applied to $W\ \ SQ$ and $FA\ \ W\ \ SQ$ schemes diverges on perturbed stretched grids with triangular elements (types $(II\ )$ $(III\ )$, and $(I\ \ )$); the method converges fast for all schemes on unperturbed grids of types (I) $(I\ \ )$. Somewhat surprisingly, in rectangular geometry, no convergence problems have been detected for the CC-NA scheme. Convergence rates of DCI for stable schemes are similar to those observed on isotropic grids (Figure 8). Figure 13 shows convergence histories on a $33\ \ 129$ grid of type $(I\ \ )$. The asymptotic rates for all converging schemes are around $0\ 5$ per iteration.



(a) Grids of type ($_p$).

(b) Grids of type ($_p$)

Figure 13. Convergence of $_1$-norms of residuals in DCI for FVD schemes with first-order drivers on stretched grids of types ($_p$) and ($_p$) with maximum aspect ratio $\mathcal{A} = 10^3$.

IX. Grids with curvature and high aspect ratio

In this section, we discuss accuracy of FVD schemes on grids with large deformations induced by a combination of curvature and high aspect ratio. The grid nodes are generated from a cylindrical mapping where $(\ \ \theta)$ denote polar coordinates with spacings of h_r and h , respectively; the innermost radius is $\ \ =\ \ $. The grid aspect ratio is defined as the ratio of mesh sizes in the circumferential and the radial directions, $\mathcal{A} = \frac{h_\theta}{h_r}$. The mesh deformation is characterized by the parameter Γ:

$$\Gamma = \frac{(1 \quad \cos(h\ ))}{h_r} \qquad \frac{h}{2h_r} = \mathcal{A}\frac{h}{2} \tag{14}$$

The following assumptions are made about the range of parameters: $\ \ 1$, $\mathcal{A} \gg 1$, and $\Gamma h_r \ \ \ 1$, which implies that both h_r and h are small. For a given value of $\mathcal{A}$, the parameter Γ may vary: $\Gamma \gg 1$ corresponds to meshes with large curvature-induced deformation; $\Gamma \ \ \ 1$ indicates meshes that are locally (almost) Cartesian. In a mesh refinement that keeps $\mathcal{A}$ fixed, $\Gamma = \ \ (\mathcal{A}h\ )$ asymptotes to zero. This property implies that on fine enough grids with fixed curvature and aspect ratio, the discretization error convergence is expected to be the same as on similar grids generated on rectangular domains with no curvature.

We focus on convergence of discretization errors on high-Γ grids with large curvature-induced deformations. Considered manufactured solutions predominantly vary in the radial direction of small mesh spacing.

Four basic types of 2D grids are studied in the cylindrical geometry. In distinction from the computational grids used in the rectangular geometry, random node perturbation is not applied to high-Γ cylindrical grids because even small perturbations in the circumferential direction may lead to non-physical control volumes.

(a) Grid of type (III).

(b) Grid of type (IV).

Figure 14. Representative $9 \quad 33$ **stretched high- grids.**

Computational grids are stretched grids with radial extent of $1 \qquad 1\ 2$ and angular extent of $20 \quad$ with a fixed maximal aspect ratio $\mathcal{A} \qquad 1\ 100$. The grids have four times more nodes in the radial direction than in the circumferential direction. The maximal value of parameter Γ changes approximately from 24 to $1\ 5$. The stretching ratio is changing as $\quad = 1\ 25\ \ 1\ 11\ \ 1\ 06\ \ 1\ 03$ and $1\ 01$. Representative stretched grids of types (III) and $(I \quad)$ are shown in Figure 14. The tests are performed for the manufactured solution $\quad = \sin(5 \quad)$.

A. Approximate mapping method

Computations and analysis reported earlier[23 25 26] conclude that the unweighted-least-square gradient approximation is zeroth order accurate on deformed grids with high Γ. To improve the accuracy of gradient reconstruction, a least-square minimization in a mapped domain is proposed. A general approximate mapping (AM) method based on the distance function has been introduced in Ref. [1].

The AM method applies the LSQ minimization in a local coordinate system, $(\xi \ \eta)$, where η is the coordinate normal to the boundary and ξ is the coordinate tangent to the boundary. The unit vector normal to the boundary, $\hat{\mathbf{n}}$, is constructed using the distance function, readily available in practical codes, as

$$\hat{\mathbf{n}} \ = (\mathbf{r} \qquad \mathbf{r} \ ) \quad \mathbf{r} \qquad \mathbf{r} \tag{15}$$

where the position of the control-volume center is denoted $\mathbf{r}$ and the position of the closest point on the boundary is denoted $\mathbf{r}$. The unit vector tangent to the boundary is denoted as $\hat{\mathbf{t}}$.

For constructing the least-square minimization at a control-volume with the center $\mathbf{r}$, the local coordinates of a stencil point $\mathbf{r}$ are defined as

$$\xi \ = (\mathbf{r} \qquad \mathbf{r} \ ) \ \ \hat{\mathbf{t}} \tag{16}$$

$$\eta \ = (s \qquad s \ ) \tag{17}$$

where s denotes the distance function of location $\mathbf{r}$. Thus the η-coordinate corresponds to the distance from the boundary and the ξ-coordinate is the projection onto the surface. The least-square minimization yields gradients in the $(\xi \ \eta)$ directions or, equivalently, through a coordinate rotation, in the $(x \ y)$ Cartesian directions.

The left and right states at a control-volume face location, say $\mathbf{r}_f$, are reconstructed using gradients in the $(\xi\ \eta)$ directions along with constructed coordinates

$$\xi_f = (\mathbf{r}_f \quad \mathbf{r}\ )\ \ \hat{\mathbf{t}} \tag{18}$$

$$\eta_f = (s_f \quad s\ ) \tag{19}$$

The coordinate s_f should be an accurate approximation to the distance function from the actual surface, reconstructed from points on the actual surface and not from the distance function computed at the interface location. A posible approximation is

$$s_f = (s_f + s_f)\ \ 2 \tag{20}$$

where, for node-centered schemes, s_f and s_f correspond to the distance function of the two nodes defining the edge, and, for cell-centered schemes, s_f and s_f correspond to the distance function of the two cell centers adjacent to the face. For cell-centered schemes, direct reconstruction using Cartesian coordinate gradients is also possible, yielding identical results for grids constructed using advancing-layer techniques. As yet, the AM method has been applied only to the cell-centered schemes.

B. Accuracy of gradient approximation

The accuracy of gradients reconstructed in the global Cartesian coordinate system for the manufactured solution $= \sin(5\quad)$ on high-Γ grids of types $(I)\quad(I\quad)$ is summarized in Table 6. Convergence of the maximum gradient errors over all control volumes is tabulated.

Only schemes using the WLSQ method are capable of accurate gradient reconstruction on irregular high-Γ grids. The NC-WLSQ scheme reconstructs accurate gradients on deformed grids of all types. All other schemes show large (1) errors on mixed-element grids of type $(I\quad)$ with $\Gamma \gg 1$. On grids of type (III), the CC-FA-WLSQ also provides accuracy for gradient reconstruction. Schemes using unweighted least-square gradient reconstruction produce large gradient errors even on regular grids.

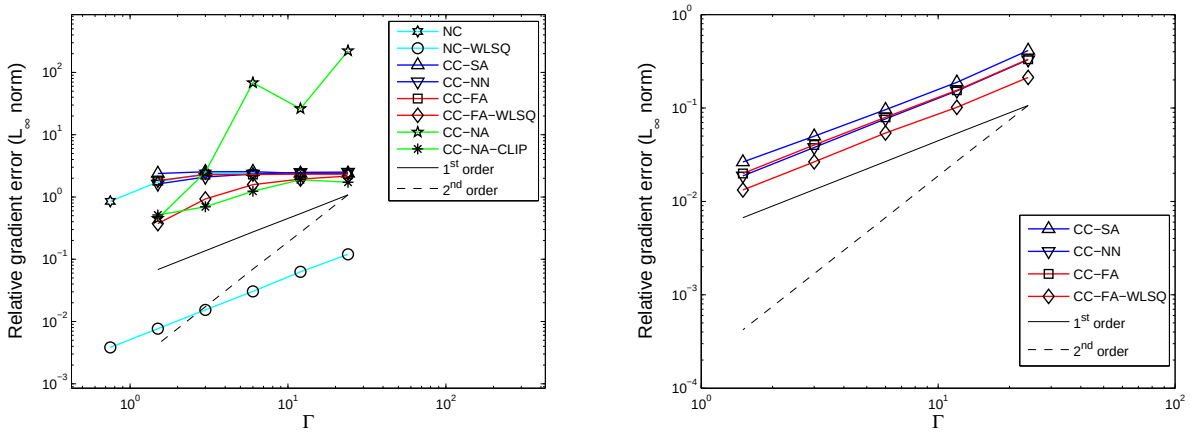
Table 6. High- grids: relative errors of gradient reconstruction in global Cartesian coordinates.

	(I)	(II)	(III)	(IV)
NC	(1)	(1)	(1)	(1)
NC-WLSQ	$(h\ )$	$(h\ )$	$(h\ )$	$(h\ )$
CC-SA	(1)	(1)	(1)	(1)
CC-NN	(1)	(1)	(1)	(1)
CC-FA	(1)	(1)	(1)	(1)
CC-FA-WLSQ	$(h\ )$	$(h\ )$	$(h\ )$	(1)
CC-NA-CLIP	$(h\ )$	$(h\ )$	(1)	(1)
CC-NA	$(h\ )$	$(h\ )$	(1)	(1)

Gradient accuracy is dramatically improved with the AM method. Table 7 shows accuracy orders for gradients reconstructed with cell-centered least-square methods in the local coordinates. All tested schemes provide accurate gradients on grids of all types. For illustration, Figure 15 shows relative accuracy of gradients reconstructed on grids of type $(I\quad)$. Note that the CC-NA scheme produces very large gradient errors. This behavior can be explained by possible node averaging degeneration on high-Γ mixed-element grids. On these grids, there are topologies where the node solution is averaged from four neighboring cells. The four cell centers involved in such averaging may be located on a straight line, thus leading to degeneration.

Table 7. High- grids: relative errors of gradient reconstruction in local AM coordinates.

	(I)	(II)	(III)	(IV)
CC-SA	$(h\ )$	$(h\ )$	$(h\ )$	$(h\ )$
CC-NN	$(h\ )$	$(h\ )$	$(h\ )$	$(h\ )$
CC-FA	$(h\ )$	$(h\ )$	$(h\ )$	$(h\ )$
CC-FA-WLSQ	$(h\ )$	$(h\ )$	$(h\ )$	$(h\ )$



(a) Cartesian coordinates

(b) Approximate mapping

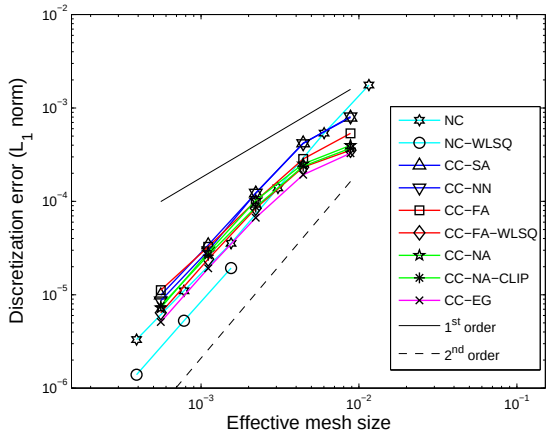
Figure 15. Convergence of relative gradient errors for FVD schemes on high- stretched grids of type ( ) with maximum aspect ratio $\mathcal{A} = 1\ 100$**.**

C. Discretization error convergence

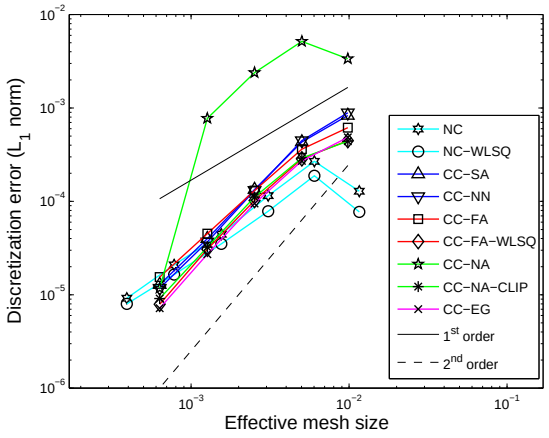
Convergence of -norms of discretization errors of FVD schemes with and without approximate mapping is shown in Figure 16. Discretization errors of the NC-WLSQ scheme in Figure 16(a) are shown only for grids with relatively low Γ; on grids with higher Γ, DCI do not converge. With the exception of the CC-NA scheme on high-Γ grids of type $(I\ )$, all other schemes show second-order convergence and very similar discretization errors. Large erratic discretization errors of the CC-NA scheme are probably caused by degeneration of the node-averaging stencil mentioned in the previous section. This explanation is supported by the evidence of accurate solutions obtained with the CC-NA scheme on low-Γ grids and on triangular grids of type (III), where such degeneration is impossible. On grids of the same size, the discretization errors of schemes using the AM method show less variation and are smaller than the errors of the corresponding schemes that do not use the AM method. The level of discretization errors obtained by the schemes with (1) error in the gradient reconstruction is not much different from the discretization error level obtained by the schemes with either the AM method (and first-order accurate gradients) or the exact gradient.

D. Convergence of defect-correction iterations

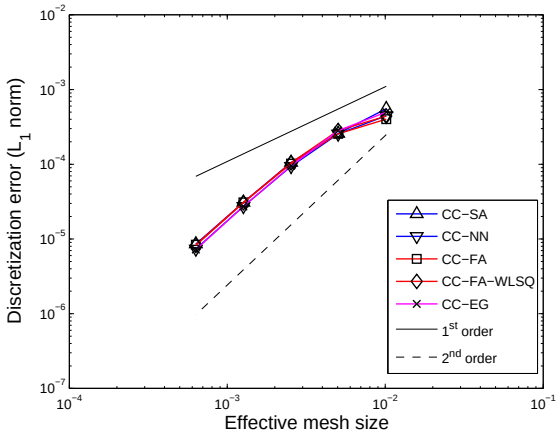
Convergence rates of DCI on irregular high-Γ grids are shown in Figure 17. The DCI method diverges for the CC-NA scheme on grids of both types and for the NC-WLSQ scheme on grids of type (III); on grids of type $(I\ )$, the NC-WLSQ scheme slowly converges. Note that for all schemes, beside the CC-SA and CC-FA schemes, convergence rates of DCI are slower than the rates on perturbed non-curved grids of similar sizes (compare Figures 13 and 17).



(a) Grids of type ( ); Cartesian coordinates



(b) Grids of type ( ); Cartesian coordinates



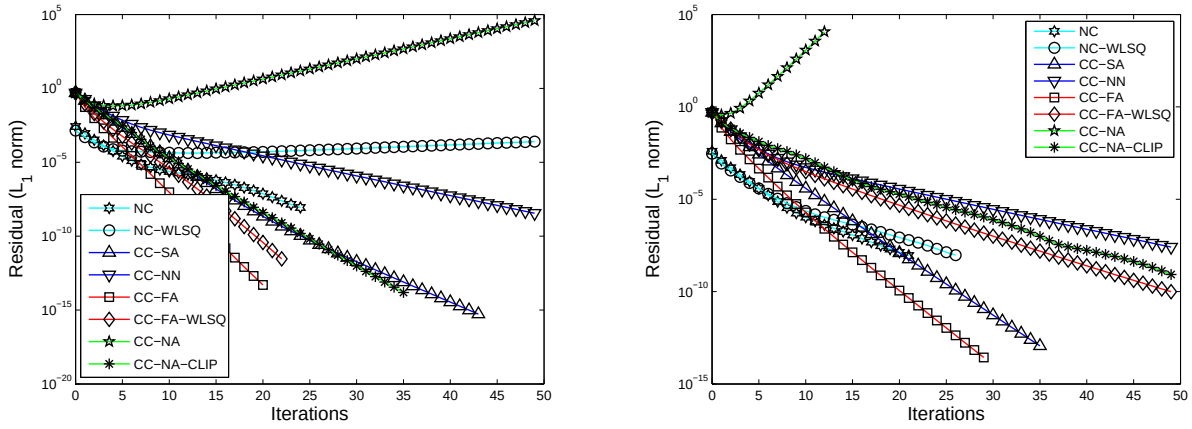
(c) Grids of type ( ); approximate mapping

Figure 16. Convergence of $_1$-norms of discretization errors of FVD schemes on high- stretched grids with maximum aspect ratio $\mathcal{A} = 1\ 100$**.**

X. Conclusions

Two node-centered and six cell-centered schemes have been compared for finite-volume discretization of a constant-coefficient convection equation as a model of the inviscid flow terms. The cell-centered nearest-neighbor (CC-NN) scheme has the lowest complexity; in particular, its stencil involves the least number of neighbors. A version of the scheme that involves smart augmentation of the least-square stencil (CC-SA) has only marginal complexity increase. All other schemes have larger complexity; the complexity of node-centered (NC) schemes are somewhat lower than complexity of cell-centered node-averaging (CC-NA) and full-augmentation (CC-FA) schemes. Defect-correction iterations (DCI) with a first-order driver is typically used for solutions of second-order finite-volume discretization (FVD) schemes. Convergence of DCI is an important consideration. The CC-NN and CC-SA schemes are promising as candidates to be iterated with full second-order linearization.

Comparisons of accuracy and convergence rates of DCI have been made for two classes of tests: the first class is representative of adaptive-grid simulations and involves irregular grids with discontinuous metrics; the second class is representative of high-Reynolds number turbulent flow simulations over a curved body. All tests have been performed



(a) Grid of type ( ).

(b) Grid of type ( )

Figure 17. Convergence of $_1$-norms of residuals in DCI for FVD schemes with first-order drivers on high- stretched grids with maximum aspect ratio $\mathcal{A} = 1\ 100$.

for smooth manufactured solutions.

For the tests of the first class performed in rectangular geometries on consistently refined grids with discontinuous metrics, the following observations have been made:

(1) Discretization errors of second-order schemes are quantitatively similar on grids with the same number of degrees of freedom. The demonstrated convergence of discretization errors closely approaches an "ideal" second-order convergence on given grids exhibited by the cell-centered scheme with exact gradients.

(2) As expected, the NC discretization errors converge with second order on triangular and regular quadrilateral grids and with first order on mixed-element (types $(I\ \ )$ and $(I\ \ )$) and perturbed quadrilateral (type $(I\ )$) grids.

(3) Discretization errors of five of the six cell-centered schemes, CC-NN, CC-SA, CC-FA, CC-FA-WLSQ, and CC-NA, converge with second order on all tested grids.

(4) The CC-NA scheme with clipping (CC-NA-CLIP) fails to approximate gradients and degrades solution accuracy to first order. The deterioration of solution accuracy is observed on very fine grids with an increased percentage of clipped nodes. On coarser grids, the accuracy of the clipped solutions is similar to the accuracy of other second-order schemes.

(5) All schemes may produce (Ah_x) large relative errors in gradient reconstruction on perturbed grids of types $(I\ )$ $(I\ \ )$; here $\mathcal{A}$ is the grid aspect ratio and h_x is the larger mesh spacing.

(6) As expected, truncation error convergence order is typically one order lower than the convergence order of corresponding discretization errors.

(7) The DCI method for FVD schemes employing weighted least-square gradient reconstruction (CC-FA-WLSQ and NC-WLSQ) diverges on perturbed stretched grids. DCI convergence rates for all other schemes, including CC-NN and CC-SA, are very fast, while slightly grid dependent; the asymptotic convergence rate is typically better than $0\ 5$ per iteration.

(8) As a recommendation for computations in geometries with no curvature, cell centered CC-NN and CC-SA schemes offer the best options of the lowest complexity and second-order discretization errors.

The tests of the second class have been performed on consistently refined stretched grids generated around a curved body, typical of those generated by the method of advancing layers. The range of grid parameters has been chosen

to enforce significant curvature-induced grid deformations, characterized by the parameter Γ. All tests have been performed for a manufactured solution smoothly varying in the radial direction.

(1) The discretization errors converge with second order and are small (approaching "ideal" second-order errors) for the CC-NN, CC-SA, and CC-FA schemes on all grids and for NC schemes on triangular grids. The errors are similar on grids with the same number of degrees of freedom. The discretization errors of the CC-NA scheme without clipping do not converge on irregular high-Γ grids.

(2) The CC-NN, CC-SA, and CC-FA schemes with least-square gradient reconstruction performed in local approximate mapping coordinates provide accurate gradients on all grids. Approximate mapping accounts for the global curvature and relies on the distance function that is typically available in practical computations. With least-square gradient reconstruction performed in global Cartesian coordinates that do not account for global curvature, only the NC-WLSQ scheme provides accurate gradients on all grids; all other schemes fail for mixed-element grids of type $(I \quad)$, generating $\quad(1)$ errors in gradient reconstruction. On grids of type (III), the only cell-centered scheme with accurate gradient is CC-FA-WLSQ scheme. Note that unweighted least-square schemes fail to approximate gradients even on regular grids of types (I) and (II). CC-NA schemes provide accurate gradients on regular grids, but exhibit poor gradient accuracy on irregular grids, even with approximate mapping.

(3) The DCI method for the CC-NA scheme without clipping diverges; for the NC-WLSQ scheme, the method either diverges or converges very slowly. Convergence rates of DCI for the CC-SA and CC-FA schemes are fast and almost grid independent; the average convergence rate is better than $0\ 5$ per iteration. The DCI convergence rates for other schemes are slower.

(4) As a recommendation for computations in curved geometries, the best option is the CC-SA scheme that offers low complexity, second-order discretization errors, and fast convergence of DCI. The CC-NN is a promising candidate to be iterated with full second-order linearization. The approximate mapping provides uniform accuracy for gradient reconstruction.

References

[1]Diskin, B., Thomas, J. L., Nielsen, E. J., Nishikawa, H., and White, J. A., "Comparison of Node-Centered and Cell-Centered Unstructured Finite-Volume Discretizations. Part I: Viscous Fluxes," AIAA Paper 2009-597, 47-th AIAA Aerospace Sciencies Meeting, Orlando, FL, January 2009, AIAA Journal in press.

[2]Thomas, J. L., Diskin, B., and Rumsey, C. L., "Towards Verification of Unstructured-Grid Solvers," *AIAA Journal*, Vol. 46, No. 12, December 2008, pp. 3070–3079.

[3]Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *J. Comp. Phys.*, Vol. 43, No. 2, 1981, pp. 357–372.

[4]Aftosmis, M., Gaitonde, D., and Tavares, T. S., "Behavior of Linear Reconstruction Techniques on Unstructured Meshes," *AIAA Journal*, Vol. 33, 1995, pp. 2038–2049.

[5]Diskin, B. and Thomas, J. L., "Accuracy Analysis for Mixed-Element Finite-Volume Discretization Schemes," NIA Report 2007-08, National Institute of Aerospace, August 2007.

[6]Holmes, D. G. and Connell, S. D., "Solution of the 2D Navier-Stokes Equations on Unstructured Adaptive Grids," AIAA Paper 89-1392, 9-th AIAA CFD conference, Washington, DC, June 1989.

[7]Rausch, R. D., Batina, J. T., and Yang, H. T., "Spatial Adaptation Procedures on Unstructured Meshes for Accurate Unsteady Aerodynamic Flow Computation," AIAA Paper 89-1392, 9-th AIAA CFD conference, Washington, DC, June 1989.

[8]Frink, N. T., "Recent Progress Toward A Three-Dimensioanl Unstructured Navier-Stokes Flow Solver," AIAA Paper 94-0061, 32-nd AIAA Aerospace Science Meeting and Exhibit, Reno, NV, January 1994.

[9]Haselbacher, A. C., "On Constrained Reconstruction Operators," AIAA Paper 2006-1274, 44-th AIAA Aerospace Science Meeting and Exhibit, Reno, NV, January 2006.

[10]Barth, T. J., "Numerical Aspects of Computing High-Reynolds Number Flow on Unstructured Meshes," AIAA Paper 91-0721, 29-th AIAA Aerospace Science Meeting, Reno, NV, January 1991.

[11]Coirier, W. J., "An Adaptively-Refined, Cartesian, Cell-Based Scheme for the Euler and Navier-Stokes Equations," NASA TM-106754, October 1994.

[12]Frink, N. T., "Tetrahedral Unstructured Navier-Stokes Methods for Turbulent Flows," *AIAA Journal*, Vol. 36, No. 11, 1998, pp. 1975–1982.

[13]Delanaye, M. and Liu, Y., "Quadratic Reconstruction Finite Volume Schemes on 3D Arbitrary Unstructured Polyhedral Grids," AIAA Paper 1999-3259, 14-th AIAA CFD Conference, Norfolk, VA, June 1999.

[14]Hirsch, C., *Numerical Computation of Internal and External Flows. Vol.1, FUndumentals of Numerical Discretization*, A Wiley-Interscience publication, John Wiley & Sons, Inc., 605 Third Avenue, New York, NY 10158-0012, USA, 1988.

[15]Syrakos, A. and Goulas, A., "Estimate of the Truncation Error of Finite Volume Discretization of the Navier-Stokes Equations on Collocated Grids," *Int. J. Numer. Meth. Fluids*, Vol. 50, 2006, pp. 103–130.

[16]Turkel, E., "Accuracy of Schemes with Nonuniform Meshes for Compressible Fluid Flows," *Applied Numerical Mathematics*, Vol. 2, 1986, pp. 529–550.

[17]Giles, M. B., "Accuracy of Node-Based Solutions on Irregular Meshes," *11-th International Conference on Numerical Methods in Fluid Dynamics*, edited by D. L. Dwoyer, M. Y. Hussaini, and R. Voigt, Lecture Notes in Physics, v. 323, Springer-Verlag, 1989, pp. 369–373.

[18]Kreiss, H. O., Manteuffel, T. A., Wendroff, B., and White, A. B., "Supra-Convergence Schemes on Irregular Grids," *Mathematics of Computations*, Vol. 47(176), October 1986, pp. 537–554.

[19]Lindquist, D. R. and Giles, M. B., "A Comparison of Numerical Schemes on Triangular and Quadrilateral Meshes," *11-th International Conference on Numerical Methods in Fluid Dynamics*, edited by D. L. Dwoyer, M. Y. Hussaini, and R. Voigt, Lecture Notes in Physics, v. 323, Springer-Verlag, 1989, pp. 273–277.

[20]Manteuffel, T. A. and White, A. B., "The Numerical Solution of the Second-Order Boundary Value Problem on Nonuniform Meshes," *Mathematics of Computations*, Vol. 47(176), October 1986, pp. 511–536.

[21]Tikhonov, A. N. and Samarsky, A. A., "Homogeneous Difference Schemes on Nonuniform Nets," *Zh. Vychsl. Mat. and Mat. Fiz.*, Vol. 2, 1962, pp. 812–832, (English translation in U.S.S.R. Comput. Math. and Math. Phys., pp. 927–953).

[22]Diskin, B. and Thomas, J. L., "Half-Space Analysis of the Defect-Correction Method for Fromm Discretization of Convection," *SIAM J. Sci. Comp.*, Vol. 22, No. 2, 2000, pp. 633–655.

[23]Smith, T. M., Barone, M. F., Bond, R. B., Lorber, A. A., and Baur, D. G., "Comparison of Reconstruction Techniques for Unstructured Mesh Vertex Centered Finite Volume Scheme," 18-th AIAA CFD Conference, Miami, FL, June 2007, AIAA Paper 2007-3958.

[24]Diskin, B. and Thomas, J. L., "Accuracy of Gradient Reconstruction on Grids with High Aspect Ratio," NIA Report 2008-12, National Institute of Aerospace, December 2008.

[25]Mavriplis, D. J., "Revisiting the Least-Square Procedure for Gradient Reconstruction on unstructured Meshes," AIAA Paper 2003-3986, 18-th AIAA CFD conference, Orlando, FL, June 2003.

[26]Petrovskaya, N. V., "The Choice of Weight Coefficients for Least-Square Gradient Approximation," *J. Math. Mod.*, Vol. 16(5), 2004, pp. 83–93, (in Russian).

AIAA JOURNAL
Vol. 48, No. 7, July 2010

Comparison of Node-Centered and Cell-Centered Unstructured Finite-Volume Discretizations: Viscous Fluxes

Boris Diskin*
National Institute of Aerospace, Hampton, Virginia 23666
James L. Thomas† and Eric J. Nielsen‡
NASA Langley Research Center, Hampton, Virginia 23681-2199
Hiroaki Nishikawa§
National Institute of Aerospace, Hampton, Virginia 23666
and
Jeffery A. White¶
NASA Langley Research Center, Hampton, Virginia 23681-2199

DOI: 10.2514/1.44940

Finite-volume discretization schemes for viscous fluxes on general grids are compared using node-centered and cell-centered approaches. The grids range from regular grids to highly irregular grids, including random perturbations of the grid nodes. Accuracy and complexity are studied for four nominally second-order accurate schemes: a node-centered scheme and three cell-centered schemes (a node-averaging scheme and two schemes using least-squares face-gradient reconstruction). The two least-squares schemes use either a nearest-neighbor or an adaptive-compact stencil at a face. The node-centered and least-squares schemes have similarly low levels of complexity. The node-averaging scheme has the highest complexity and can fail to converge to the exact solution when clipping of the node-averaged values is used. On highly anisotropic grids, typical of those encountered in grid adaptation, the least-squares schemes, the node-averaging scheme without clipping, and the node-centered scheme demonstrate similar second-order accuracies per degree of freedom. On anisotropic grids over a curved body, typical of turbulent flow simulations, the node-centered scheme is second-order accurate. The node-averaging scheme may degenerate on mixed-element grids. The least-squares schemes have to be amended to maintain second-order accuracy by either introducing a local approximate mapping or modifying the stencil to reflect the direction of strong coupling. Overall, the accuracies of the node-centered and the best cell-centered schemes are comparable at an equivalent number of degrees of freedom on isotropic and curved anisotropic grids. On stretched, randomly perturbed grids in a rectangular geometry, both gradient and discretization errors for all schemes are orders of magnitude higher than corresponding errors on regular grids.

Nomenclature

$\mathcal{A}$	=	aspect ratio
E_d	=	discretization error
E_t	=	truncation error
E_{rel}	=	relative gradient error
$\mathbf{e}_{\alpha\beta}$	=	unit vector in $[\alpha\beta]$ direction
$\mathbf{e}^{\perp}$	=	vector normal to the vector $\mathbf{e}$
f	=	forcing function
f^h	=	discrete approximation to the forcing function
h_e	=	effective mesh size, L_1 norm of $\sqrt{V}$
h_r, h_θ	=	radial and circumferential mesh spacing, respectively
h_x, h_y	=	Cartesian mesh sizes in the x and y directions, respectively
$\hat{h}_y$	=	minimal mesh spacing on stretched grids
$\{i^T\}$	=	set of nodes of cell T
$\{k_j\}$	=	set of nodes connected to node j by edges
N	=	total number of mesh points
N_x, N_y	=	number of grid points in the x and y directions, respectively
$\mathbf{n}$	=	outward-directed area vector
$\hat{\mathbf{n}}$	=	outward unit normal vector
$\breve{\mathbf{n}}_i$	=	inward-directed area vector of a face opposite to node i
R	=	radius of curvature
$\mathbf{r}$	=	coordinate vector
r, θ	=	polar coordinates
s	=	distance to the designated boundary
T	=	triangle or tetrahedron
$\{T_j\}$	=	set of triangles/tetrahedra around node j
U	=	exact solution of Poisson's equation
U^h	=	discrete solution of Poisson's equation
$\overline{\nabla U}$	=	gradient of solution U evaluated by Green–Gauss formula
$\widehat{\nabla U}$	=	gradient of solution U evaluated by least-squares method
V	=	measure of a control volume
x, y	=	Cartesian coordinates
β	=	stretching factor
Γ	=	curvature-induced grid deformation parameter
Δ	=	Laplace operator
$\partial^e U$	=	edge derivative of solution U

Presented as Paper 597 at the 47th AIAA Aerospace Sciences Meeting, Orlando, FL, 5–8 January 2007; received 15 April 2009; revision received 16 February 2010; accepted for publication 17 February 2010. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0001-1452/10 and $10.00 in correspondence with the CCC.

*100 Exploration Way; University of Virginia, Department of Mechanical and Aerospace Engineering, Charlottesville, Virginia 22904; bdiskin@nianet.org, Senior Member AIAA.

†Computational AeroSciences Branch, Mail Stop 128; James.L.Thomas@nasa.gov. Fellow AIAA.

‡Computational AeroSciences Branch, Mail Stop 128; Eric.J.Nielsen@nasa.gov. Senior Member AIAA.

§100 Exploration Way; hiro@nianet.org. Member AIAA.

¶Computational AeroSciences Branch, Mail Stop 128; Jeffery.A.White@nasa.gov. Senior Member AIAA.

$\partial^f U$ = face derivative of solution U
θ^L, θ^R = angles between edges in two dimensions
μ = edge median
ξ, η = local coordinates
ρ = random number $\rho \in [-1, 1]$
Ω, $\partial\Omega$ = control volume and control-volume boundary, respectively
$|\cdot|$ = absolute value of a scalar or a vector
$\|\cdot\|$ = norm of interest (e.g., L_1 or L_∞)
∇ = gradient operator
∇_r = reconstructed gradient

Subscript

p = grid with perturbed nodes

Superscripts

L, R = triangles to the left and right of an edge

I. Introduction

BOTH node-centered (NC) and cell-centered (CC) finite-volume discretizations (FVDs) are widely used for complex three-dimensional (3-D) turbulent simulations in aerospace applications. The relative advantages of the two approaches have been extensively studied in the search for methods that are accurate, efficient, and robust over the broadest possible range of grid and solution parameters. The topic was discussed in a panel session at the 2007 AIAA Computational Fluid Dynamics (CFD) Conference, but a consensus did not emerge. One of the difficulties in assessing the two approaches is that comparative calculations were not completed in a controlled environment (i.e., computations were made with different codes and different degrees of freedom), and the exact solutions were not known.

In this paper, a subset of the discretization elements needed in turbulent simulations, namely that of the viscous discretization, is compared in a controlled environment. In particular, Poisson's equation is considered as a model of viscous discretization. The method of manufactured solution is used, so that the exact solution is known and smooth on the scale of the grids. Theoretical and computational studies of accuracy and complexity are conducted for a range of grids.

The two-dimensional (2-D) grids considered range from structured (regular) grids to irregular grids composed of arbitrary mixtures of triangles and quadrilaterals. Highly irregular grids are deliberately constructed through random perturbations of structured grids to bring out the worst possible behavior of the solution. Two classes of tests are considered. The first class of tests involves both isotropic and highly anisotropic grids, typical of those encountered in grid adaptation. The second class of tests involves grids varying strongly anisotropically over a curved body, typical of those encountered in high-Reynolds-number turbulent flow simulations.

Four nominally second-order accurate schemes, a NC scheme and three CC schemes, are compared for computational complexity and gradient and discretization errors at equivalent degrees of freedom. The CC schemes include a node-averaging (CC–NA) scheme and two least-squares face-gradient reconstruction schemes differing in their stencils: a nearest-neighbor (CC–NN) stencil and an adaptive-compact stencil (CC–CS). The effect of clipping is studied for the CC–NA scheme. The current version of the CC–CS scheme is derived for triangular grids, but it can be formally applied to quadrilateral and mixed-element grids, for which it is similar to the CC–NN scheme. It is expected that an effective mixed-element version of the CC–CS scheme can be derived, but it is not currently available. For the second class of tests, an approximately mapped (AM) least-squares approach is introduced to accommodate curved high-aspect-ratio grids. The mapping employs the distance function commonly available in practical codes and can be used with any scheme.

II. Grid Terminology

This paper studies FVD schemes for viscous fluxes on grids that are loosely defined as irregular. There is no commonly accepted definition for irregular grids and so, for clarity, this section specifies the grid terminology used in the paper.

A grid is classified as periodic if it has 1) a periodic node connectivity pattern (i.e., the number of edges per node changes periodically) and 2) a periodic cell distribution (i.e., the grid is composed of periodically repeated combinations of cells). Thus, periodic grids can be analyzed by Fourier analysis. Grids that are derived from periodic grids by a smooth mapping are called regular grids. Regular grids include, but are not limited to, grids derived from Cartesian ones, triangular grids obtained by diagonal splitting with a periodic pattern, smoothly stretched grids, skewed grids, smooth curvilinear grids, etc. Grids that cannot be smoothly mapped to a periodic grid are called irregular grids. Grids with varying local topology are called unstructured (e.g., grids with the number of edges changing from node to node with no pattern).

The regular and irregular grids considered in this paper are derived from an underlying (possibly mapped) Cartesian grid with mesh sizes h_x and h_y and the aspect ratio $\mathcal{A} = h_x/h_y$; both mesh sizes of the underlying grid are assumed to be small, $h_y \ll 1$ and $h_x \ll 1$. Irregularities are introduced locally and do not affect grid topology and metrics outside of a few neighboring cells. A local grid perturbation is called random if it is independent of local perturbations introduced beyond some immediate neighborhood. For computational grids generated for the reported studies, grid irregularities are introduced in two ways (both local and random): 1) the quadrilateral cells of the underlying grid are randomly split (or not split) into triangles and 2) the grid nodes are perturbed from their original positions by random shifts, taken as fractions of the local mesh size.

Four basic grid types are considered:

1) Type I consists of regular quadrilateral (i.e., mapped Cartesian) grids.

2) Type II consists of regular structured triangular grids derived from the regular quadrilateral grids by the same diagonal splitting of each quadrilateral.

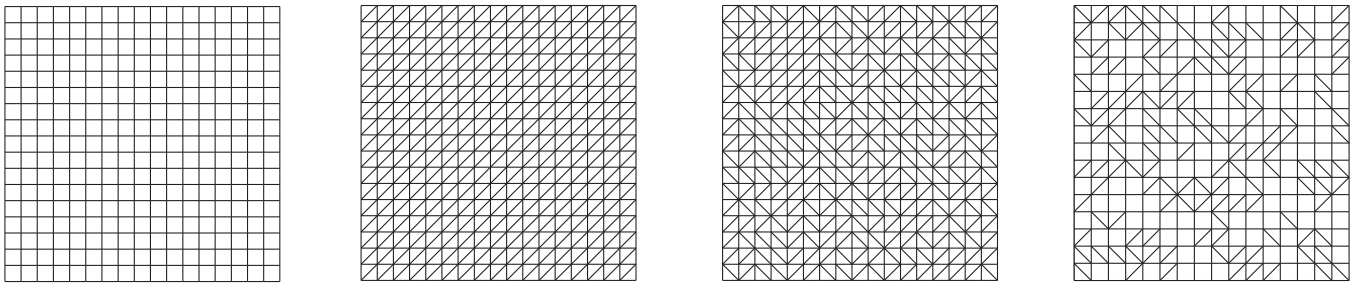
3) Type III consists of random triangular grids, in which regular quadrilaterals are split by randomly chosen diagonals, each diagonal orientation occurring with a probability of half.

4) Type IV consists of random mixed-element grids, in which regular quadrilaterals are randomly split or not split by randomly chosen diagonals, the probabilities of splitting and of choosing a particular diagonal are half.

Grids of types III–IV are irregular and unstructured because there is no periodic connectivity pattern. Nodes of any basic-type grid can be perturbed from their initial positions by random shifts, thus leading to four additional perturbed grid types that are designated by subscript p as I_p–IV_p. All perturbed grids are irregular, because there is no periodic cell distribution. The representative grids are shown in Fig. 1.

Our main interest is the accuracy and complexity of FVD schemes on general irregular grids with a minimum set of constraints. In particular, grid smoothness is not required, neither on individual grids nor in the limit of grid refinement. The only major requirement for a sequence of refined grids is to satisfy the consistent refinement property. This property requires the maximum distance across the grid cells to decrease consistently with the increase of the total number of grid points, N. In particular, the maximum distance should tend to zero as $N^{-1/2}$ in 2-D computations. For 3-D unstructured grids, the consistent refinement property is studied in [1]. On 2-D grids, the effective mesh size h_e is computed as the L_1 norm of the square root of the control volumes.

The locations of discrete solutions are called data points. For consistency with the 3-D terminology, the 2-D cell boundaries are called faces, and the term edge refers to a line (possibly virtual) connecting the neighboring data points. Each face is characterized by two vectors: 1) the edge vector, which connects the data points of the cells sharing the face and 2) the directed-area vector, which is normal

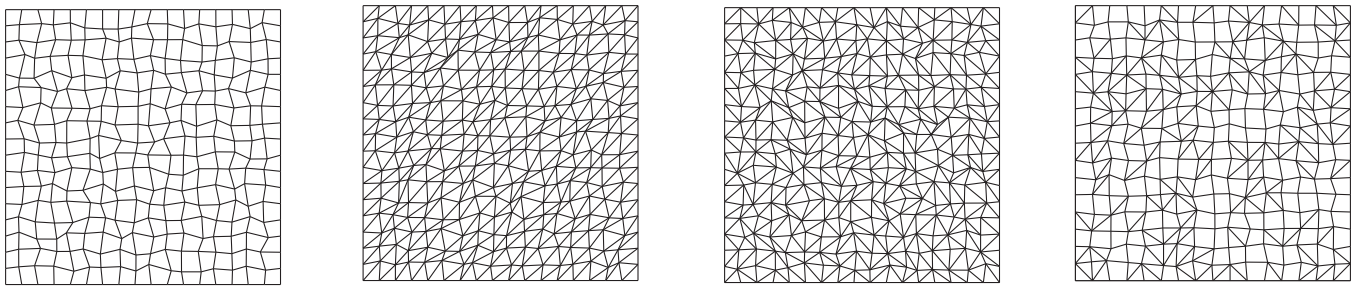


a) Type *I*: regular quadrilateral grid

b) Type *II*: regular structured triangular grid

c) Type *III*: random triangular grid

d) Type *IV*: random mixed grid



e) Type I_p: perturbed quadrilateral grid

f) Type II_p: perturbed structured triangular grid

g) Type III_p: perturbed random triangular grid

h) Type IV_p: perturbed random mixed grid

Fig. 1 Grids: a) and b) typical regular, and c)–h) irregular.

to the face with magnitude equal to the face area. For each cell/face combination, the vectors are directed outward.

For grids of types I_p–IV_p, the random node perturbation in each dimension is defined as $\frac{1}{4}\rho h$, where $\rho \in [-1,\ 1]$ is a random number, and h is the local mesh size along the given dimension. With these perturbations, triangular cells in the rectangular geometry can approach zero volume. The random perturbations are introduced independently on all grids, implying that on grids of types I_p–IV_p, the ratios of neighboring cell volumes and face areas are random and do not approach unity in the limit of grid refinement.

III. Finite-Volume Discretization Schemes

The considered model problem is Poisson's equation,

$$\Delta U = f \tag{1}$$

subject to Dirichlet boundary conditions, where function f is a forcing function. The 2-D primal meshes generated for this study are composed of triangular and quadrilateral cells. The FVD schemes are derived from the integral conservation law,

$$\oint_{\partial\Omega} \nabla U \cdot \hat{\mathbf{n}}\ \mathrm{d}s = \int_{\Omega} f\,\mathrm{d}\Omega \tag{2}$$

where ∇U is the solution gradient, Ω is a control volume with boundary $\partial\Omega$, and $\hat{\mathbf{n}}$ is the outward unit normal vector. The general FVD approach requires partitioning the domain into a set of nonoverlapping control volumes and numerically implementing Eq. (2) over each control volume.

CC discretizations assume solutions are defined at the centers of the primal-grid cells, with the primal cells serving as the control volumes. The cell center is typically defined as the average of the vortices defining the cell (i.e., not necessarily a centroid). NC discretizations assume solutions are defined at the primal-mesh nodes. For NC schemes, control volumes are constructed around the mesh nodes by the median-dual partition: the centers of primal cells are connected with the midpoints of the surrounding faces. These nonoverlapping control volumes cover the entire computational domain and compose a mesh that is dual to the primal mesh. Both CC and NC control-volume partitions are illustrated in Fig. 2.

A. Cell-Centered Finite-Volume Discretization Schemes

In CC discretizations, the conservation law in Eq. (2) is enforced on control volumes that are primary cells. The flux at a face is computed as the inner product of the solution gradient at the face and the directed-area vector. The at-face solution gradient is typically reconstructed from the solution values at the neighboring cells and augmented with the edge-directional gradient. Augmentation is used to decrease the scheme susceptibility to odd–even decoupling [2,3]. Two possible augmentation strategies, edge normal and face tangent, are discussed in [2,4]. In this paper, the face-tangent augmentation strategy is implemented for CC schemes. The schematic of the face-tangent gradient augmentation is illustrated in Fig. 3.

With reference to Fig. 2, the gradient, $\nabla_r U_{04}$ (at the face-linking nodes 0 and 4) is computed as

$$\nabla_r U_{04} = \frac{1}{\hat{\mathbf{n}} \cdot \hat{\mathbf{e}}}\,\partial^e U \hat{\mathbf{n}} + \partial^f U \left[\hat{\mathbf{f}} - \frac{\hat{\mathbf{f}} \cdot \hat{\mathbf{e}}}{\hat{\mathbf{n}} \cdot \hat{\mathbf{e}}}\,\hat{\mathbf{n}}\right] \tag{3}$$

Here,

$$\hat{\mathbf{e}} = (\mathbf{r}_B - \mathbf{r}_A)/|\mathbf{r}_B - \mathbf{r}_A| \tag{4}$$

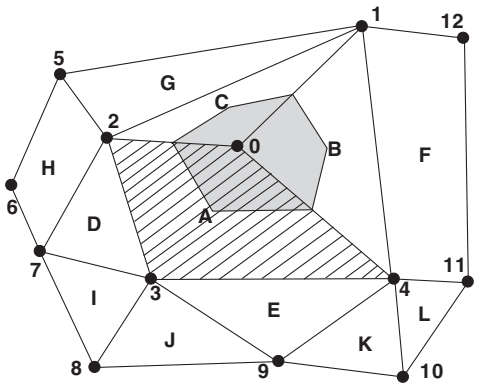


Fig. 2 Control-volume partitions for FVDs. Numbers 0–12 and letters A–L denote grid nodes and primal cell centers, respectively. The control volume for a NC discretization around grid node 0 is shaded. The control volume for a CC discretization around the cell center *A* is hashed.

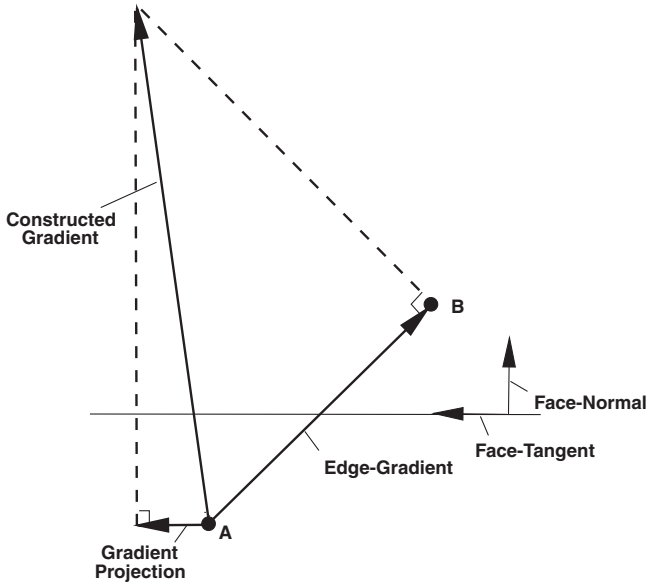


Fig. 3 Face-tangent gradient augmentation; gradient projection is $\partial^f U \hat{\mathbf{f}}$.

is the unit vector aligned with the virtual edge $[A,\ B]$, $\mathbf{r}_A$ and $\mathbf{r}_B$ are the cell-center coordinate vectors, $\hat{\mathbf{n}}$ is the unit vector normal to the control-volume face $[0, 4]$ directed outward from cell center A,

$$\hat{\mathbf{f}} = (\mathbf{r}_0 - \mathbf{r}_4)/|\mathbf{r}_0 - \mathbf{r}_4| \tag{5}$$

is a unit vector normal to $\hat{\mathbf{n}}$,

$$\partial^e U = \frac{U_B - U_A}{|\mathbf{r}_B - \mathbf{r}_A|} \tag{6}$$

is the edge-directional derivative, and $\partial^f U$ is the solution derivative computed along the face $[0, 4]$.

The face-tangent augmentation enforces that $\nabla_r U_{04}$ recovers: 1) the edge-directional derivative,

$$\nabla_r U_{04} \cdot \hat{\mathbf{e}} = \partial^e U \tag{7}$$

and 2) the face-tangent derivative,

$$\nabla_r U_{04} \cdot \hat{\mathbf{f}} = \partial^f U \tag{8}$$

The CC FVD schemes considered in this paper differ only in computing $\partial^f U$.

1. Node-Averaging Face Gradient

In the CC–NA schemes, the solution derivative along the face, $\partial^f U$, is computed as the divided difference between the solution values reconstructed at the nodes from the surrounding cell centers. With respect to Fig. 2, the solution at node 0 is reconstructed by averaging solutions defined at the cell centers A, B, and C. The solution reconstruction proposed in [5,6] and used in [7] is an averaging procedure that is based on a constrained optimization to satisfy some Laplacian properties. The scheme is second-order accurate and stable when the coefficients of the introduced pseudo-Laplacian operator are close to one. It has been shown in [8] that this averaging procedure is equivalent to an unweighted least-squares linear fit. For the face $[0, 4]$,

$$\partial^f U = \frac{\hat{U}_0 - \hat{U}_4}{|\mathbf{r}_0 - \mathbf{r}_4|} \tag{9}$$

where $\hat{U}_i$ and $\mathbf{r}_i$ are the averaged solution and the coordinate vector of the node i.

On highly stretched and deformed grids, some coefficients of the pseudo-Laplacian may become negative or larger than two, which has a detrimental effect on stability and robustness [9,10]. Holmes and Connell [5] proposed to enforce stability by clipping the coefficients between 0 and 2. The CC–NA schemes with clipping represent a current standard in practical CFD for applications involving CC finite-volume formulations [11]. As shown further in the paper, clipping seriously degrades the solution accuracy.

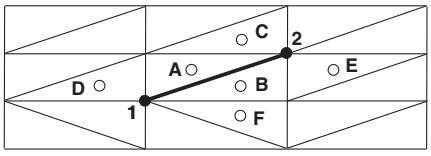
2. Least-Squares Scheme Face Gradient

An alternative CC scheme relies on a face-based least-squares method. First, an auxiliary face gradient $\widehat{\nabla U}$ is reconstructed within a face using a least-squares procedure. Then, the derivative along the face is computed as

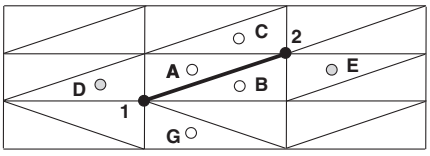
$$\partial^f U = \widehat{\nabla U} \cdot \hat{\mathbf{f}} \tag{10}$$

The two approaches to determine stencils for the least-squares linear fit at a face are described as follows. The CC–NN six-point stencil consists of the two prime cells sharing the face and their face neighbors, which share one of the face nodes. In Fig. 4a, the CC–NN stencil for the highlighted face is denoted by circles.

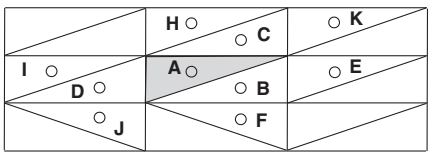
The CC–CS is important for discretizations on high-aspect-ratio grids of types II and III to correctly represent the direction of the strong coupling. It is constructed by choosing between two stencils for face least-squares gradient reconstruction: a six-point stencil and a minimal (typically four-point) stencil. In general, the minimal stencil takes advantage of the local topology associated with grids generated with advancing layer methods, and it is intended for long faces of high-aspect-ratio triangular grids.



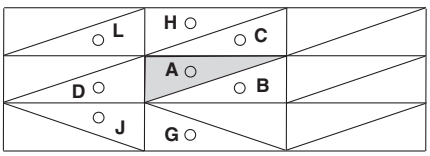
a) Face least-squares stencil with CC-NN scheme



b) Face least-squares stencil with CC-CS scheme



c) Laplacian stencil for the shaded cell with CC-NN scheme



d) Laplacian stencil for the shaded cell with CC-CS scheme

Fig. 4 Stencils on high-aspect-ratio grids of type III. Figures are vertically expanded for better visualization.

Specifically, at each face, the CC–CS method first attempts to construct a six-point stencil by combining two prime cells and four auxiliary cells; each auxiliary cell is associated with a prime cell and a face node. The method chooses the auxiliary cell that 1) shares the face node, 2) is located on the opposite side of the face from the associated prime cell center, 3) is not already in the stencil, and 4) has the shortest distance to the center of the associated prime cell. The six-point stencil for the highlighted diagonal face is denoted by the union of empty and filled circles in Fig. 4b. Note that cell F in the CC–NN stencil (Fig. 4a) is replaced by cell G in the six-point CC–CS stencil. For the prime cell A on high-aspect-ratio grids, the nearest cell that shares node 1 and is on the opposite side of the face $[1, 2]$ is cell$G$, not cell$F$. In the process of construction, the closest auxiliary cell associated with each primal cell is identified. The minimal stencil is defined by the union of the prime cells and their closest associated auxiliary cells. In Fig. 4b, cell G is the closest auxiliary cell to the primal cell A, cell C is the closest auxiliary cell to the primal cell B, and the minimal stencil is shown as empty circles. Note that, in some local geometries, a prime cell may have no auxiliary cells. In such cases, the minimal stencil consists of less than four points.

The CC–CS method selects the minimal stencil if either the six-point stencil cannot be formed following the rules 1–4 (which may happen next to the boundaries or in curved geometries) or the minimal stencil represents an ideal four-point pairwise construction. The four-point pairwise construction is considered ideal if one can form two pairs, with each pair satisfying the three following geometrical conditions. The data points within the pair 1) are on opposite sides of the face, 2) are closer than a predefined threshold (typically taken as a fraction of the larger local mesh size), and 3) have a skew angle (the angle between the vector connecting the points and the face directed-area vector) smaller than a predefined threshold. For computations on high-aspect-ratio grids, the distance threshold has been chosen as $\frac{3}{16}h_x$, where h_x is the larger mesh size of the background Cartesian grid, and the skew threshold has been chosen as $\sin^{-1}(0.1)$. The four-point stencil in Fig. 4b is considered ideal.

Figures 4c and 4d compare CC–NN and CC–CS stencils corresponding to the FVD of Poisson's equation on the shaded cell. The CC–CS scheme uses minimal stencils for diagonal and horizontal faces and a six-point stencil for vertical faces. The CC–CS stencil is more compact than the CC–NN stencil and provides a three-point vertical structure centered at the shaded cell center that better reflects the grid anisotropy direction.

Remark: It is known that on high-aspect-ratio curved grids, unweighted least-squares methods have difficulties with reconstructing accurate gradients within a cell [12–14]. Inverse distance weighting has been shown to improve gradient accuracy. For face-centered least-squares reconstruction, the usual weightings (with distances measured from the face center) do not improve gradient accuracy, because all points involved in least-squares stencils are typically at comparable distances from the face center. A modified weighting, which is based on minimal distances from the two cell centers across the face, with an extended stencil (the stencil that is used in CC–NA scheme) improves gradient accuracy on high-aspect-ratio curved grids derived by an advanced-layer method. The weighting effectively reduces the extended stencil to the minimal stencil of the CC–CS scheme. However, the method led to unstable formulations on general irregular grids and was not pursued further.

B. Node-Centered Finite-Volume Discretization Scheme

The second-order accurate NC FVD scheme illustrated by Fig. 5 represents a standard CFD approach to NC viscous discretizations. The scheme approximates the integral flux through the dual faces adjacent to the edge $[0, 4]$ as

$$\int_{A\mu B} \nabla U \cdot \hat{\mathbf{n}}\, \mathrm{d}s \approx \nabla_r U_{A\mu} \cdot \mathbf{n}_{A\mu} + \nabla_r U_{\mu B} \cdot \mathbf{n}_{\mu B} \tag{11}$$

where μ is the median of the edge $[0, 4]$. The gradient is reconstructed separately at each dual face as follows.

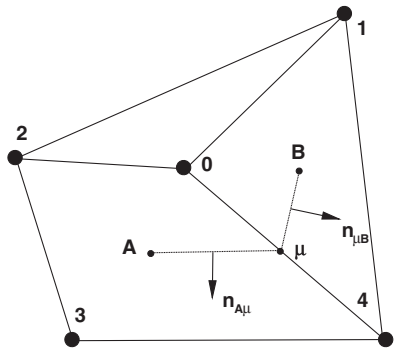


Fig. 5 Illustration of gradient reconstruction for viscous terms on mixed grids with median-dual partition.

For the triangular element contribution, the gradient is determined from a Green–Gauss evaluation at the primal-grid element:

$$\nabla_r U_{\mu B} = \overline{\nabla U}_{014} \tag{12}$$

The gradient overbar denotes a gradient evaluated by the Green–Gauss formula on the primal cell identified by the point subscripts. With fully triangular elements, the formulation is equivalent to a Galerkin finite-element scheme with a linear basis function [9,15]. Analysis in Appendix A shows that on unperturbed triangular grids of types II and III in rectangular geometries, the formulation recovers the five-point Laplacian stencil of the type I grids, independent of aspect ratio.

For the quadrilateral element contribution, the gradient $\nabla_r U_{A\mu}$ is constructed as the Green–Gauss gradient augmented with the edge derivative,

$$\nabla_r U_{A\mu} = \overline{\nabla U}_{0234} + [\partial^e U - \overline{\nabla U}_{0234} \cdot \mathbf{e}_{04}]\mathbf{e}_{04} \tag{13}$$

where

$$\partial^e U = \frac{U_4 - U_0}{|\mathbf{r}_4 - \mathbf{r}_0|} \tag{14}$$

is the edge derivative, U_i is the solution at node i, and

$$\mathbf{e}_{04} = \frac{\mathbf{r}_4 - \mathbf{r}_0}{|\mathbf{r}_4 - \mathbf{r}_0|} \tag{15}$$

is the unit vector aligned with the edge $[0, 4]$. The edge-normal augmentation illustrated in Fig. 6 is used to enforce that the constructed gradient recovers 1) the edge-directional derivative,

$$\nabla_r U_{A\mu} \cdot \mathbf{e}_{04} = \partial^e U \tag{16}$$

and 2) the Green–Gauss gradient projected on the direction normal to $\mathbf{e}_{04}$:

$$\nabla_r U_{A\mu} \cdot \mathbf{e}_{04}^{\perp} = \overline{\nabla U}_{0234} \cdot \mathbf{e}_{04}^{\perp} \tag{17}$$

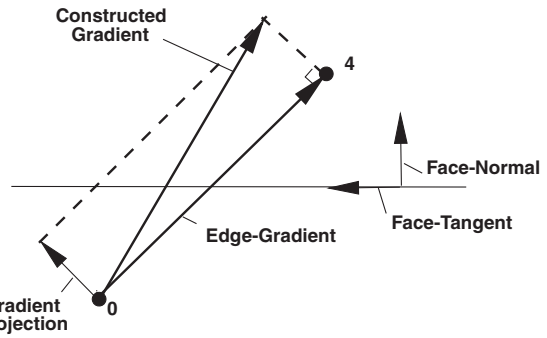


Fig. 6 Edge-normal gradient augmentation; gradient projection is $\overline{\nabla U} - (\overline{\nabla U} \cdot \mathbf{e})\mathbf{e}$.

Note that, for grids with dual faces perpendicular to the edges, the edge gradient $\partial^e U$ is the only contributor. It has been shown [1,16] that the scheme possesses second-order accuracy for viscous fluxes on general isotropic mixed-element grids.

IV. Complexity of Discretization Stencils

The size of the stencil for the viscous discretization is examined for 2-D and 3-D CC and NC FVD schemes. Estimates are made for Cartesian meshes split into triangular and tetrahedral elements, neglecting any boundary effects.

In 2-D, two splittings of the Cartesian grid are considered. The first splits each quadrilateral cell with a diagonal oriented in the same direction. The second splits the cells with diagonals of face-adjacent quadrilaterals oriented in the opposite direction. The second splitting is slightly more analogous to the 3-D splitting. In 3-D, half of the grid nodes have 18 incident edges (32 incident tetrahedra) and half have six incident edges (eight incident tetrahedra). Each of the tetrahedra interior to an originally hexahedral cell is defined by four nodes, each with 18 incident edges. Each of the four surrounding tetrahedra within an originally hexahedral cell is defined by three nodes with 18 incident edges and one node with six incident edges.

Table 1 shows stencil-size estimates for triangular/tetrahedral grids and a numerical calculation on an actual 3-D turbulent viscous grid that includes boundary effects. There is a slight difference in the 2-D estimates from the two splittings (entries separated by slashes in the table), depending on the diagonalization pattern. The CC–NA stencil is the largest. The CC–NN stencil is only slightly larger than the stencil of the NC discretization, in both estimation and computation. The complexity of the CC–CS stencil is even smaller.

V. Analysis Methods

The accuracy of FVD schemes is analyzed for known exact or manufactured solutions. The forcing function and boundary values are found by substituting this solution into the Poisson equation with Dirichlet boundary conditions. The discrete forcing function is defined at the data points.

A. Discretization Error

The main accuracy measure is the discretization error E_d, which is defined as the difference between the exact discrete solution U^h of the discretized Eq. (2) and the exact continuous solution U to the differential Eq. (1),

$$E_d = U - U^h \tag{18}$$

where U is sampled at the data points.

B. Truncation Error

Another accuracy measure commonly used in computations is truncation error. Truncation error E_t characterizes the accuracy of approximating the differential equation (1). For finite differences, it is defined as the residual obtained after substituting the exact solution U into the discretized differential equations [17]. For FVD schemes, the traditional truncation error is usually defined from the time-dependent standpoint [18,19]. In the steady-state limit, it is defined (e.g., in [20]) as the residual computed after substituting U into the normalized discrete Eq. (2),

Table 1 Average size of the viscous stencil on triangular (2-D) and tetrahedral (3-D) grids. The two numbers for CC 2-D schemes correspond to different diagonalization patterns

	NC	CC–NA	CC–NN
2-D estimate	7	13/16	10/9
3-D estimate	13	79	15
3-D numerical	14	69	15

$$E_t = \frac{1}{V}\left[-\int_\Omega f^h \,\mathrm{d}\Omega + \oint_{\partial\Omega} (\nabla_r U \cdot \hat{\mathbf{n}})\,\mathrm{d}s\right] \tag{19}$$

where V is the measure of the control volume,

$$V = \int_\Omega \mathrm{d}\Omega \tag{20}$$

f^h is an approximation of the forcing function f on Ω, and the integrals are computed according to some quadrature formulas. Note that convergence of truncation errors is expected to show the order property only on regular grids. It has been long known that, on irregular grids, the design-order discretization-error convergence can be achieved. even when truncation errors exhibit a lower-order convergence or, in some cases, do not converge at all [21–23].

C. Accuracy of Gradient Reconstruction

Yet another important accuracy measure is the accuracy of gradient approximation at a control-volume face. For second-order convergence of discretization errors, the gradient is usually required to be approximated with at least first order. For each face, accuracy of the gradient is evaluated by comparing the reconstructed gradient $\nabla_r U$ with the exact gradient ∇U computed at the face center. The gradient reconstruction uses a discrete representation (usually injection) of the exact solution U at data points on a given grid. The accuracy of gradient reconstruction is measured as the relative gradient error,

$$E_{\mathrm{rel}} = \frac{\|\epsilon\|}{\|G\|} \tag{21}$$

where functions ϵ and G define at-face magnitudes of the gradient error and the exact gradient, respectively,

$$\epsilon = |\nabla_r U - \nabla U| \tag{22}$$

and

$$G = |\nabla U|$$

and $\|\cdot\|$ is a norm of interest computed over the entire computational domain. For the NC scheme, the exact and reconstructed gradients are evaluated at the centers of primal cells.

VI. Isotropic Irregular Grids

A. Grid Refinement

A sequence of consistently refined grids of type III_p is generated on the unit square $[0,\ 1] \times [0,\ 1]$. Irregularities are introduced at each grid independently. The ratio of areas of neighboring faces can be as large as $3\sqrt{2}$. The ratio of the neighboring volumes can be arbitrarily high, because a control volume can be arbitrarily small. Isotropic grids randomly generated for this study have 0.01% of cell volumes smaller than $\frac{1}{10}\frac{1}{N}$, where N is the total number of grids nodes.

B. Gradient Reconstruction Accuracy

The accuracy of gradient reconstruction for isotropic irregular grids is first order for all methods [24], which is sufficient for second-order discretization accuracy. As an example, the gradient reconstruction tests are performed for the manufactured solution $U = \sin(\pi x + 2\pi y)$. Figure 7 shows convergence of the L_∞ norms of relative gradient errors computed on a sequence of refined grids of type III_p. All methods provide first-order gradient approximations and very similar relative errors. Note that, because the gradients of the NC scheme are evaluated at the primal cell centers, the effective mesh size of gradient reconstruction is the same for all schemes.

C. Convergence of Truncation and Discretization Error

The numerical tests evaluating convergence of truncation and discretization errors are performed with Dirichlet boundary conditions specified from the manufactured solution $U = \sin(\pi x + 2\pi y)$.

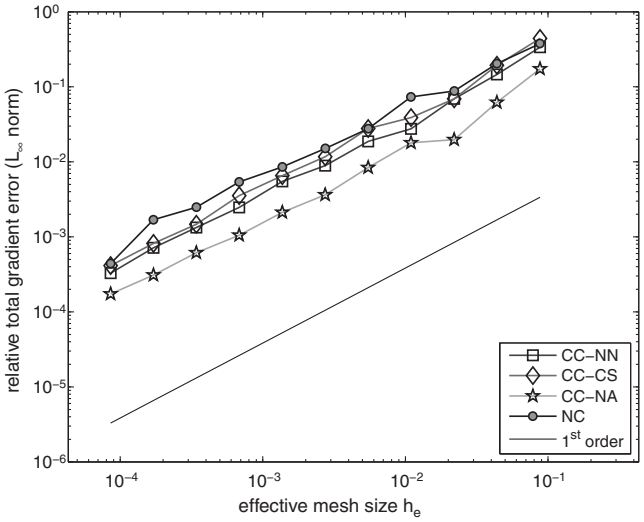
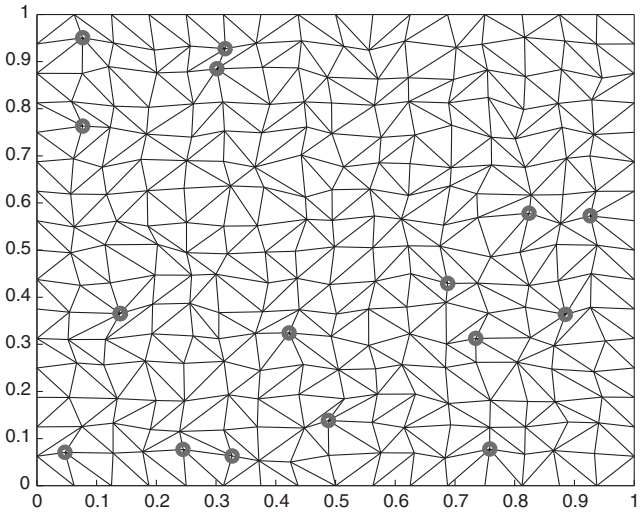
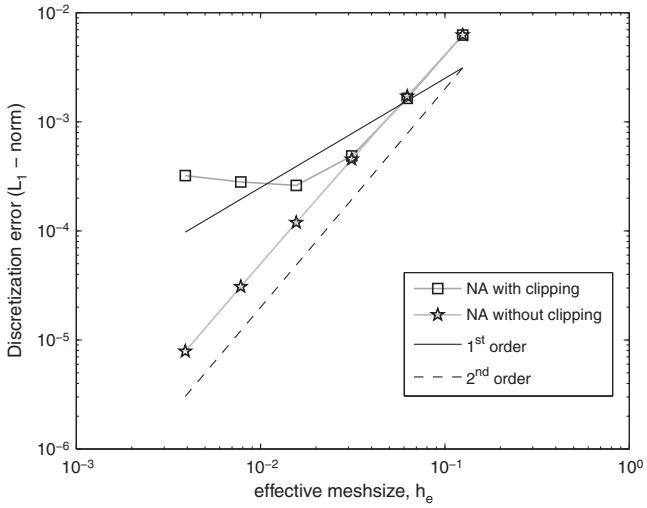


Fig. 7 Accuracy of gradient reconstruction on isotropic irregular grids of type III$_p$.



a) Random triangular grid with 17^2 nodes. Clipped nodes are circled



b) Discretization errors

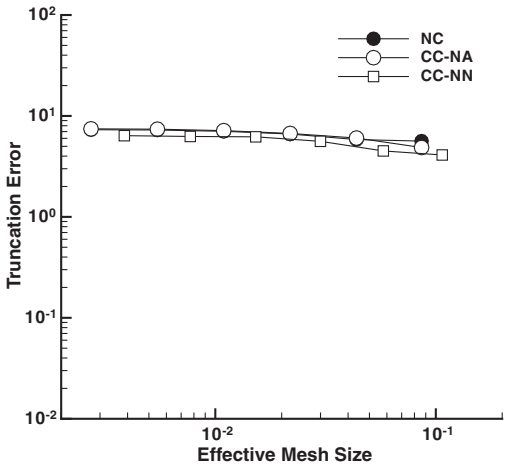
Fig. 9 Accuracy of CC–NA schemes on isotropic irregular triangular grids of type III$_p$.

For CC formulations, the solution is specified on all cells linked to the boundary. Figure 8 shows convergence of the L_1 norms of truncation and discretization errors for the NC and two CC formulations on grids of type III$_p$. As predicted in [1,16], truncation errors do not converge on irregular grids in any norm. Discretization errors converge with second order for all formulations considered. The discretization errors of the CC and NC FVD schemes are almost overplotted, indicating a similar accuracy per degree of freedom. Note that a given multidimensional grid typically has more primal cells than nodes. Thus, on a given grid, a CC scheme has more degrees of freedom than a NC scheme and, consequently, is expected to have a better accuracy.

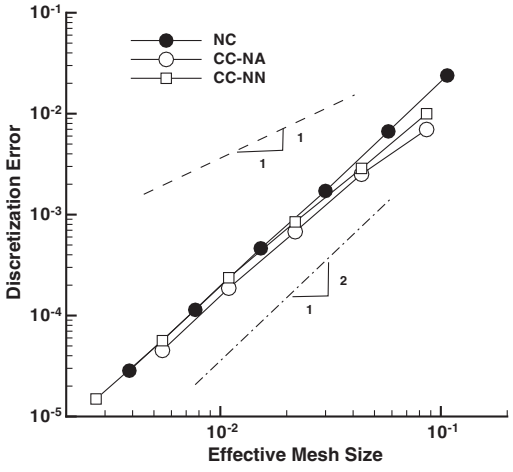
D. Effects of Clipping

The tests reported in this section are performed for the CC–NA schemes and demonstrate detrimental effects of clipping on accuracy of gradient approximation and on the discretization accuracy. The accuracy is evaluated for the manufactured solution $U = \sin(2\pi y)$. Considered irregular grids of type III$_p$ are derived from underlying isotropic (unit aspect ratio) Cartesian grids covering the unit square. Figure 9a shows an example of an isotropic random triangular grid of type III$_p$ with 17^2 nodes. About 7% of the interior nodes are clipped.

It has been demonstrated in [25] that the face gradients computed by the CC–NA scheme with clipping do not approximate the exact gradients on grids of type III$_p$. The normal and tangential components of the computed gradients were evaluated within interior faces and compared with the exact gradient components at the face center. The maximum norms of the deviations between the computed and the exact gradient components did not converge in grid refinement. The CC–NA scheme without clipping provided a first-order-accurate



a) Truncation Errors



b) Discretization Errors

Fig. 8 Convergence of L_1 norms of truncation and discretization errors on random triangular grids of type III$_p$.

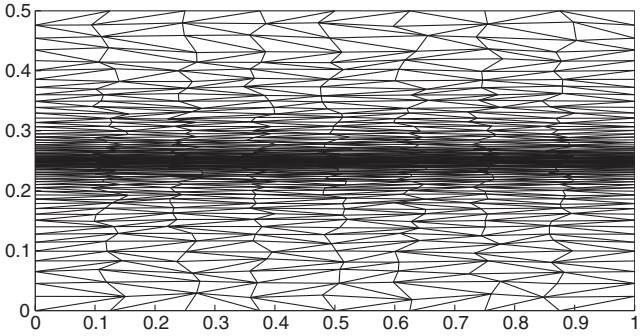


Fig. 10 Stretched grid of type III$_p$ with 9 × 65 nodes.

gradient approximation. Figure 9b exhibits convergence of the L_1 norms of discretization errors. Although the CC–NA scheme without clipping demonstrates second-order convergence on all grids, convergence of the CC–NA scheme with clipping degrades to zeroth order on finer grids. Although not shown, the L_∞ norms of the discretization errors converge with the same orders as the corresponding L_1 norms.

VII. Anisotropic Grids

This section considers FVD schemes on irregular stretched grids generated on rectangular domains. Figure 10 shows an example grid with the maximal aspect ratio $\mathcal{A} = 1000$. A sequence of consistently refined stretched grids is generated on the rectangle $(x, y) \in [0, 1] \times [0, 0.5]$ in the following three steps.

1) A background regular rectangular grid with $N = (N_x + 1) \times (N_y + 1)$ nodes and the horizontal mesh spacing $h_x = 1/N_x$ is stretched toward the horizontal line $y = 0.25$. The y coordinates of the horizontal grid lines in the top half of the domain are defined as

$$y_{(N_y/1)+1} = 0.25; \qquad y_j = y_{j-1} + \hat{h}_y \beta^{j-[(N_y/2)+1)]};$$
$$j = \frac{N_y}{2} + 2, \ \ldots, N_y, N_y + 1 \tag{23}$$

Here, $\hat{h}_y = h_x/A$ is the minimal mesh spacing between the vertical lines, $\mathcal{A} = 1000$ is a fixed maximal aspect ratio, and β is a stretching factor that is found from the condition $y_{N_y+1} = 1$. The stretching in the bottom half of the domain is defined analogously.

2) Irregularities are introduced by random shifts of interior nodes in the vertical and horizontal directions. The vertical shift is defined as $\Delta y_j = \frac{3}{16}\rho \min(h_y^{j-1}, h_y^j)$, where ρ is a random number between -1 and 1, and h_y^{j-1} and h_y^j are vertical mesh spacings on the background stretched mesh around the grid node. The horizontal shift is introduced analogously, $\Delta x_i = \frac{3}{16}\rho h_x$. With these random node perturbations, all perturbed quadrilateral cells are convex.

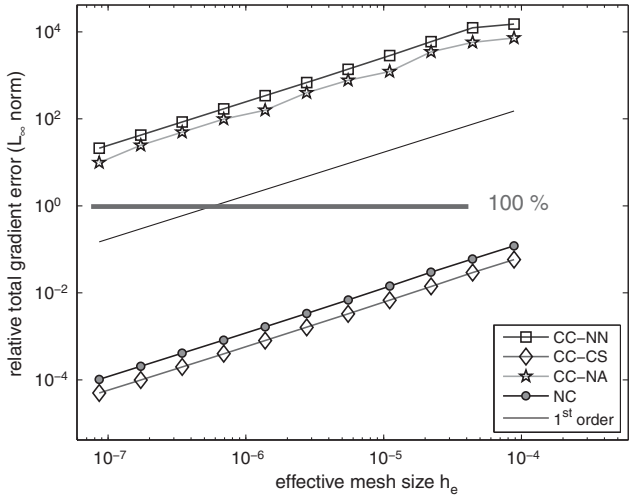
3. Each perturbed quadrilateral is randomly triangulated with one of the two diagonal choices; each choice occurs with a probability of one half.

A recent study [24] assessed the accuracy of gradient approximation on various irregular grids with a high aspect ratio of $\mathcal{A} = h_y/h_x \gg 1$. The study indicates that, for rectangular geometries and functions predominantly varying in the direction of small mesh spacing (y direction), gradient reconstruction is accurate. For manufactured solutions significantly varying in the direction of larger mesh spacing (x direction), the face-gradient reconstruction may produce extremely large $O(\mathcal{A}h_x)$ relative errors affecting the accuracy of the y-directional gradient component. Figures 11a and 11b confirm this analysis and show examples of gradient approximations that exhibit first-order accuracy and large relative errors on high-aspect-ratio grids of type III. On these grids, the NC scheme and CC–CS scheme produce accurate gradients for all solutions, independent of grid aspect ratio. Accuracy of gradients reconstructed with CC–NN and CC–NA schemes is directly proportional to $\mathcal{A}h_x$ and typically poor for solutions varying in the x direction of larger mesh spacing, unless the grids are extremely fine. For solutions varying predominantly in the y direction of smaller mesh spacing, all schemes produce accurate gradients.

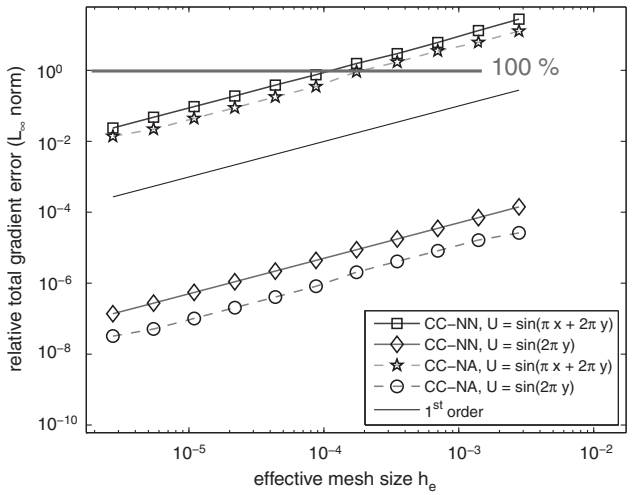
A summary of the previous results [24] for grids of all types (supplemented by the results for the CC–CS scheme) is presented in Table 2. All considered gradient reconstruction methods are accurate on regular quadrilateral grids of type I, but they may generate large relative errors on irregular grids of types I$_p$–IV$_p$ with perturbed nodes. The CC–NA and CC–NN methods may also have large relative errors on unperturbed grids of types II–IV. The CC–CS gradients are accurate for unperturbed triangular grids; the accuracy of CC–CS gradients is similar to the accuracy of the CC–NN gradients on mixed-element grids of type IV. The NC method using the Green–Gauss approach always provides accurate gradients on unperturbed grids.

However, a poor gradient reconstruction accuracy does not necessarily imply a large discretization error. Mavriplis [12] reported (second-order) accurate NC solutions, even on grids with large gradient reconstruction errors. Here, similar results are observed for CC and NC formulations.

Sequences of consistently refined stretched grids with a maximum aspect ratio of $\mathcal{A} = 1000$, including 9×65, 17×129, 33×257, and 65×513 nodes have been considered. The corresponding stretching ratios are $\beta \approx 1.207$, 1.098, 1.048, and 1.025. The grids of types III and III$_p$ are representative for general perturbed and unperturbed grids, respectively. Convergence of the L_1 norms of discretization



a) $U = \sin(\pi x + 2\pi y)$; $\mathcal{A} = 10^6$



b) $U = \sin(\pi x + 2\pi y)$ and $U = \sin(2\pi y)$, $\mathcal{A} = 10^3$

Fig. 11 Relative errors in approximation of face gradients on anisotropic grids of type III.

Table 2 Relative error of gradient reconstruction on anisotropic grids in rectangular domains

Grids	I	II	III	IV	I_p–IV_p
NC	$O(h_x^2)$	$O(h_x)$	$O(h_x)$	$O(h_x)$	$O(\mathcal{A}h_x)$
CC–NA	$O(h_x^2)$	$O(\mathcal{A}h_x^2)$	$O(\mathcal{A}h_x)$	$O(\mathcal{A}h_x)$	$O(\mathcal{A}h_x)$
CC–NN	$O(h_x^2)$	$O(\mathcal{A}h_x^2)$	$O(\mathcal{A}h_x)$	$O(\mathcal{A}h_x)$	$O(\mathcal{A}h_x)$
CC–CS	$O(h_x^2)$	$O(h_x^2)$	$O(h_x)$	$O(\mathcal{A}h_x)$	$O(\mathcal{A}h_x)$

errors for the manufactured solution $U = \cos(\pi x + 2\pi y)$ is shown in Fig. 12. The highly stretched grids are not well suited with the manufactured solution, but such a mismatch is chosen intentionally to demonstrate convergence in the worst-case scenario.

All tests have been performed stochastically [i.e., multiple grids (ten)] with different irregularities; patterns have been independently generated on each scale (same number of nodes). The plot symbols indicate the mean errors, and the bars indicate the maximum and minimum errors observed on each scale. The effective mesh size is practically the same for all CC schemes at a given scale, but for visualization purposes, plots of the CC–NA and CC–CS schemes are shifted to the right and the left, respectively, of the CC–NN scheme.

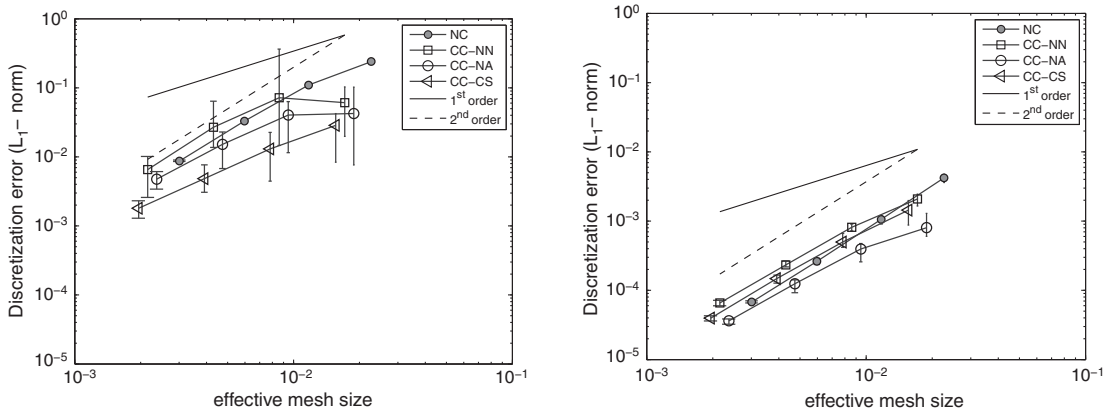
All discretization errors are relatively small and converge with second order. The errors on grids of type III are about two orders of magnitude smaller than the errors on the grids of type III_p. The NC scheme is remarkably insensitive to grid irregularities on all grids. Large variations of discretization errors are observed for CC schemes on coarse grids of type III_p. The largest variation is with the CC–NN scheme. Error variations for all schemes are decreasing on finer scales. On grids of type III, the error variations are small on all scales. The CC schemes tend to show smaller errors on coarser grids, but they require finer grids to establish the second-order convergence. Although not shown, on grids of type III_p, the level of errors for the solution $U = \cos(2\pi y)$ varying only in the y direction is more than two orders of magnitude smaller than the level of errors for the solution $U = \cos(\pi x + 2\pi y)$ that has a significant variation in the x direction.

VIII. Grids with Curvature and High-Aspect Ratio

This section discusses the accuracy of FVD schemes on grids with large deformations induced by a combination of curvature and a high aspect ratio. Grids of types I–IV are considered for the cylindrical geometry. Random node perturbation is not applied, because even small perturbations in the circumferential direction may lead to nonphysical control volumes. Representative stretched grids of types III and IV are shown in Fig. 13. The grid nodes are generated from a cylindrical mapping, where (r, θ) denotes polar coordinates with spacings of h_r and h_θ, respectively. The innermost radius is $r = R$. The grid aspect ratio is defined as the ratio of mesh sizes in the circumferential and the radial directions, $\mathcal{A} = Rh_\theta/h_r$. The mesh deformation is characterized by the parameter Γ:

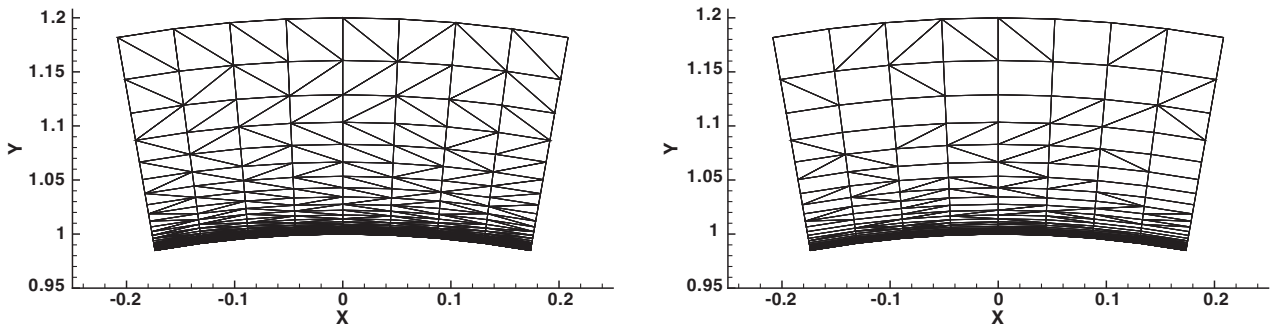
$$\Gamma = \frac{R[1 - \cos(h_\theta)]}{h_r} \approx \frac{Rh_\theta^2}{2h_r} = \mathcal{A}\frac{h_\theta}{2} \tag{24}$$

The following assumptions are made about the range of parameters: $R \approx 1, \mathcal{A} \gg 1$, and $\Gamma h_r \ll 1$, which implies that both h_r and h_θ are small. For a given value of $\mathcal{A}$, the parameter Γ may vary: $\Gamma \gg 1$ corresponds to meshes with large curvature-induced deformation, and $\Gamma \ll 1$ indicates meshes that are locally (almost) Cartesian. In a mesh refinement that keeps $\mathcal{A}$ fixed, $\Gamma = O(\mathcal{A}h_\theta)$ asymptotes to zero. This property implies that, on fine enough grids with a fixed curvature and an aspect ratio, the discretization-error convergence is expected to be the same as on similar grids generated on rectangular domains with no curvature.



a) Type III_p b) Type III

Fig. 12 Convergence of discretization errors for solution $U = \cos(\pi x + 2\pi y)$ on stretched grids with a maximum aspect ratio of $\mathcal{A} = 1000$.



a) Grid of type III b) Grid of type IV

Fig. 13 Representative 9 × 33 stretched high-Γ grids.

The focus in this section is on convergence of discretization errors on high-Γ grids with large curvature-induced deformations, following a previous study [24] that focused on gradient accuracy. The considered manufactured solutions predominantly vary in the radial direction of small mesh spacing.

A. Accuracy of Gradient Approximation

Gradient approximation accuracy on deformed grids with high Γ has been studied in the literature, mostly in regard to NC discretizations of inviscid terms [12–14]. The observations and analysis indicated that the unweighted least-squares methods poorly approximate gradients at control-volume centers. The main reasons for poor gradient approximation are 1) the stencil deformation and 2) heavy reliance of the unweighted least-squares method on solutions at distant points. Weighted least-squares methods have been proposed to reduce the effect of distant points and, thus, to improve gradient accuracy.

The situation is different for the viscous terms, for which the gradient reconstruction is required at the control-volume face, not at the center. The gradients of the NC scheme and the gradients of the CC–CS scheme on triangular grids use the minimal stencil and are expected to be accurate on unperturbed grids, independent of aspect ratio. For other CC schemes, the at-face gradient reconstruction is more difficult. The more extended stencils of least-squares methods involved either in CC–NA or in CC–NN gradient reconstruction are significantly deformed, and reconstructions generate large errors. Weighted least-squares methods are not effective, because all distances from stencil points to the face center are similar.

To improve the accuracy of gradient reconstruction, a general approximate mapping (AM) method is proposed. The AM method is motivated by the observation that, in an exactly mapped coordinate system (e.g., in polar coordinates for grids generated around a circle), gradient approximation for a radial function is as good as the gradient approximation in domains with no curvature. The AM method described next is a second-order approximation to the exact mapping.

The AM method constructs a local mapping based on the distance function that supplies the distance from a field point to designated boundaries and is readily available in practical codes. In this paper, we use the exact distance function defined at the cell centers. A more practical alternative (not used here) is to define the distance function at the grid nodes. The least-squares minimization is applied in a local coordinate system (ξ, η), where η is the coordinate normal to the boundary, and ξ is the coordinate parallel to the boundary. Figure 14 illustrates construction of the local coordinates. The vector normal to the boundary is constructed at the face center μ as an average of two normal vectors defined at the cell centers across the face. The corresponding unit vector $\hat{\mathbf{n}}_\mu$ is defined as

$$\hat{\mathbf{n}}_\mu = \frac{\mathbf{r}_A - \mathbf{r}_A^* + \mathbf{r}_B - \mathbf{r}_B^*}{|\mathbf{r}_A - \mathbf{r}_A^* + \mathbf{r}_B - \mathbf{r}_B^*|} \tag{25}$$

where $\mathbf{r}_A$ and $\mathbf{r}_B$ are the positions of the control-volume centers, and $\mathbf{r}_A^*$ and $\mathbf{r}_B^*$ are the corresponding positions of the closest boundary points. The distance to the boundary at the face center μ is approximated as

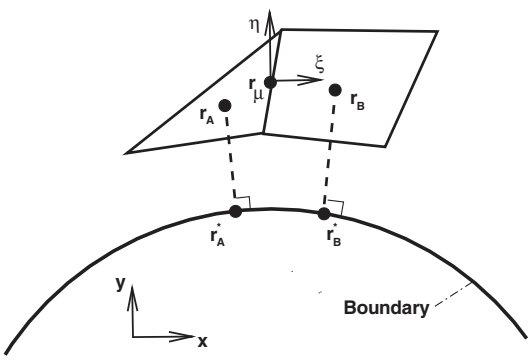


Fig. 14 Sketch of coordinate system used in AM method.

Table 3 High-Γ grids: relative errors of gradient reconstruction

	I	II	III	IV
NC	$O(h_\theta^2)$	$O(h_\theta)$	$O(h_\theta)$	$O(h_\theta)$
CC–NN	$O(h_\theta^2)$	$O(1)$	$O(1)$	$O(1)$
CC–NN–AM	$O(h_\theta^2)$	$O(h_\theta^2)$	$O(h_\theta)$	$O(h_\theta)$
CC–CS	$O(h_\theta^2)$	$O(h_\theta^2)$	$O(h_\theta^2)$	$O(1)$
CC–NA	$O(h_\theta^2)$	$O(h_\theta^2)$	$O(\mathcal{A}h_\theta)$	$O(\mathcal{A}h_\theta)$
CC–NA–AM	$O(h_\theta^2)$	$O(h_\theta^2)$	$O(h_\theta)$	$O(h_\theta)$

$$s_\mu = \frac{|\mathbf{r}_A - \mathbf{r}_A^*| + |\mathbf{r}_B - \mathbf{r}_B^*|}{2} \tag{26}$$

The unit vector normal to $\hat{\mathbf{n}}_\mu$ is denoted as $\hat{\tau}_\mu$. For constructing the least-squares minimization at a control-volume face with the center $\mathbf{r}_\mu$, each stencil point P is mapped onto the local coordinates $(\xi_P,\ \eta_P)$ by

$$\xi_P = (\mathbf{r}_P - \mathbf{r}_\mu) \cdot \hat{\tau}_\mu \tag{27}$$

$$\eta_P = s_P - s_\mu \tag{28}$$

where $s_P = |\mathbf{r}_P - \mathbf{r}_P^*|$.

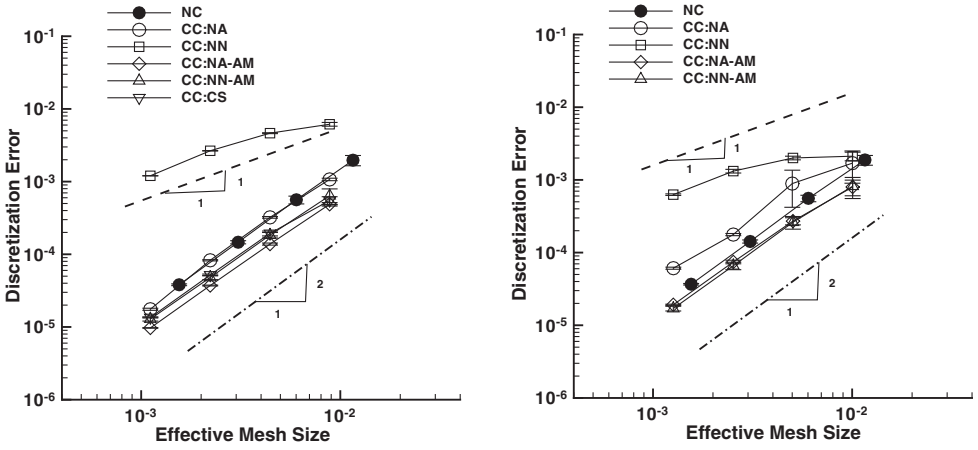
The gradient approximation accuracy for a radial function on high-Γ grids of types I–IV from the previous study [24], supplemented with the CC–CS and CC–NA–AM results, is summarized in Table 3. Convergence of the maximum gradient errors over all faces is tabulated. Note that large $O(\mathcal{A}h_\theta)$ relative errors for the CC–NA scheme occur on high-Γ grids of type III at only the radially oriented faces in the gradient component tangential to the face; the errors at other faces and in the gradient component normal to the radial face are small.

B. Discretization-Error Convergence

Discretization errors of CC schemes are compared with the errors of the NC scheme on refined stretched high-Γ grids of types III and IV. The tests are performed for the manufactured solution $U = \sin(5\pi r)$. The computational grids (see Fig. 13) are derived from background regular cylindrical grids with a radial extent of $1 \le r \le 1.2$ and an angular extent of 20 deg. The background grids have four times more nodes in the radial direction than in the circumferential direction. The grid-refinement study is performed on grids stretched in the radial direction, with a fixed maximal aspect ratio of $\mathcal{A} \approx 1000$. The maximal value of parameter Γ changes approximately from 24 to 3. The stretching ratio is changing as $\beta = 1.25$, 1.11, 1.06, and 1.03.

Convergence of the L_1 norms of the discretization errors on grids of type III is shown in Fig. 15a. All tests have been performed stochastically. The plot symbols again indicate the mean errors, and bars indicate the maximum and minimum errors observed on each scale. As expected, error variations observed on grids of the same scale due to stochastic grid irregularities are small for all schemes and decreasing for smaller scales (larger number of degrees of freedom). The errors of the NC, CC–NA, CC–CS, CC–NN–AM, and CC–NA–AM solutions converge with second order and are almost overplotted on fine grids, indicating the same accuracy per degree of freedom. The errors of the CC–NN scheme are significantly higher and converge with first order.

Convergence of the L_1 norms of the discretization errors on grids of type IV, shown in Fig. 15b, is similar to the results in Fig. 15a. The effective mesh sizes of CC and NC formulations are much closer on mixed grids than on triangular grids. The CC–CS scheme is omitted because, on mixed-element grids, its current version is similar to the CC–NN scheme. Note also that the CC–NA scheme may lose stability on high-Γ mixed-element grids. On these grids, there are topologies for which the node solution is averaged from four neighboring cells. The four cell centers involved in such averaging may be located on a straight line, thus leading to degeneration. In



a) Convergence on grids of type III b) Convergence on grids of type IV

Fig. 15 Convergence of the L_1 norms of the discretization errors on high-Γ grids.

these (rare) instances, large negative contributions appear on the main diagonals of the full linearization matrix. The scheme may still be solved and even provide a reasonable accuracy. The AM version of the CC–NA scheme, CC–NA–AM, is always stable. Overall, discretization errors of the NC scheme and the best CC schemes (CC–CS, CC–NN–AM, and CC–NA–AM) converge with second order, are insensitive to grid irregularities, and are comparable at an equivalent number of degrees of freedom.

IX. Conclusions

Complexity and accuracy of NC and CC FVDs have been compared for Poisson's equation as a model of viscous fluxes. Considering complexity, the NC scheme has the lowest complexity (i.e., its stencil involves the least number of degrees of freedom). The CC schemes using least-squares face-gradient reconstruction, the CC–NN and the CC–CS schemes, have complexity comparable with that of the NC scheme. Complexity of the CC–NA scheme is the highest.

The accuracy comparisons have been made for two classes of tests. The first class is representative of adaptive-grid simulations and involves irregular grids in rectangular geometries. The second class is representative of high-Reynolds number turbulent flow simulations over a curved body and involves highly stretched grids, typical of those generated by the method of advancing layers. All tests have been performed for smooth manufactured solutions on consistently refined grids. Grid perturbations and stretching have been intentionally introduced independently of solution variation to bring out the worst possible behavior.

For the tests of the first class, only the CC–NA scheme with clipping can fail to approximate gradients and/or to converge to the exact solution. However, note that the clipping is introduced mainly for stability of the inviscid solution and can be avoided for the viscous terms. All other schemes demonstrate similar qualities:

1) The discretization errors converge with second order and are quantitatively similar on grids of the same type with equivalent degrees of freedom. On high-aspect-ratio randomly perturbed grids, discretization errors for all schemes are orders of magnitude higher than corresponding errors on unperturbed grids.

2) Gradient reconstruction may produce $O(\mathcal{A}h_x)$ large relative errors on grids of types I_p–IV_p, where $\mathcal{A}$ is the grid aspect ratio and h_x is the larger mesh spacing.

3) Truncation errors do not converge, as expected.

For the tests of the second class, the range of grid parameters has been chosen to enforce significant curvature-induced grid deformations, characterized by parameter Γ. These high-Γ tests proved to be more discriminating:

1) The discretization errors are small and converge with second order for the NC scheme, for approximate mapping schemes (CC–NN–AM and CC–NA–AM), for the CC–NA scheme, and for the CC–CS scheme on triangular grids. The CC–NN scheme without approximate mapping shows first-order convergence and the highest level of discretization errors.

2) Accurate gradient reconstruction is provided by the NC scheme and the CC–NN–AM and CC–NA–AM schemes on all grids and by the CC–CS scheme on triangular grids. On high-Γ grids of types II–IV, the CC–NN scheme without approximate mapping generates $O(1)$ errors in gradient reconstruction. The CC–NA scheme may produce large relative gradient errors proportional to the product of the grid aspect ratio and the larger mesh spacing.

3) Without AM, the CC–NA scheme may degenerate on mixed grids.

The major conclusion is that the accuracy and complexity of the NC and the best CC schemes on irregular grids are comparable at equivalent number of degrees of freedom.

X. Acknowledgment

This paper was supported by the NASA Fundamental Aeronautics Program, Supersonics Project, NASA Research Announcement contract NNL07AA23C, with N. K. Yamaleev as the principal investigator.

Appendix A: Node-Centered Discretization on Grids of Types II and III in Rectangular Geometries

In this section, we show that the NC discretization of the Laplacian is equivalent to the standard finite-difference formula for arbitrary aspect-ratio grids of types II and III in rectangular geometry. Consider a set $\{T_j\}$ of triangles/tetrahedra that share a node j. For the NC scheme, the Green–Gauss gradient within each cell is given by

$$\overline{\nabla U}^{\,T} = \frac{1}{D\Omega^T} \sum_{i \in \{i^T\}} U_i \check{\mathbf{n}}_i \tag{A1}$$

where D is the number of spatial dimensions, $D = 2$ for triangles, $D = 3$ for tetrahedra, Ω^T is the volume of cell T, $\{i^T\}$ is a set of nodes of the cell T, and $\check{\mathbf{n}}_i$ is the inward-directed area vector of the face opposite to the node i. Then, the NC discretization (or equivalently, the standard Galerkin discretization) of the Laplacian at j is defined as

$$\int_\Omega \Delta U \, \mathrm{d}\Omega = \int_{\partial\Omega} \nabla U \cdot \mathbf{n} = -\sum_{T \in \{T_j\}} \frac{1}{D^2 \Omega^T} \sum_{i \in \{i^T\}} U_i (\check{\mathbf{n}}_i \cdot \check{\mathbf{n}}_j^T) \tag{A2}$$

where Ω is the dual control volume around j and $\check{\mathbf{n}}_j^T$ is the inward-directed area vector opposite to node j in cell T. The right-hand side of Eq. (A2) can be separated into two terms:

$$\int_\Omega \Delta U \,\mathrm{d}\Omega = -\frac{1}{D^2}\sum_{T\in\{T_j\}}\left(\frac{\check{\mathbf{n}}_j^T\cdot\check{\mathbf{n}}_j^T}{\Omega^T}\right)U_j - \frac{1}{D^2}\sum_{T\in\{T_j\}}\sum_{i\in\{i^T\},i\neq j}\left(\frac{\check{\mathbf{n}}_i\cdot\check{\mathbf{n}}_j^T}{\Omega^T}\right)U_i \tag{A3}$$

The first term contains contributions from the node value U_j, and the second term contains contributions from the neighbors.

For general 2-D triangular grids (Fig. A1),

$$\int_\Omega \Delta U \,\mathrm{d}\Omega = -\frac{1}{4}\sum_{T\in\{T_j\}}\left(\frac{\check{\mathbf{n}}_j^T\cdot\check{\mathbf{n}}_j^T}{\Omega^T}\right)U_j - \frac{1}{4}\sum_{k\in\{k_j\}}\left(\frac{\check{\mathbf{n}}_k^L\cdot\check{\mathbf{n}}_j^L}{\Omega^L}+\frac{\check{\mathbf{n}}_k^R\cdot\check{\mathbf{n}}_j^R}{\Omega^R}\right)U_k \tag{A4}$$

where $\{k_j\}$ is a set of neighbors of j, and the normals are inward normals, as defined in Fig. A1. This can be written also in terms of angles between edges,

$$\int_\Omega \Delta U \,\mathrm{d}\Omega = -\frac{1}{4}\sum_{T\in\{T_j\}}\left(\frac{\check{\mathbf{n}}_j^T\cdot\check{\mathbf{n}}_j^T}{\Omega^T}\right)U_j + \frac{1}{2}\sum_{k\in\{k_j\}}(\cot\theta^L+\cot\theta^R)U_k \tag{A5}$$

which is often used to show that the discretization is positive for triangulations with $\theta^L+\theta^R<\pi$. Consider now a grid of type III, shown in Fig. A2, which is constructed by inserting diagonals into a Cartesian grid. For this particular diagonal splitting, node 3 does not contribute to the discretization equation (A4), because it is not a neighbor to node j, and nodes 1, 5, and 7 do not contribute, because the angles θ^L and θ^R are both 90 deg; therefore, the coefficient $(\cot\theta^L+\cot\theta^R)$ vanishes. This is, in fact, true for any diagonal splittings: contributions from the corner nodes 1, 3, 5, and 7 are always zero, either because it is not in the actual stencil or because the coefficient vanishes. Observe also that angles θ^L and θ^R for other nodes are independent of the diagonal splitting; thus, we always have

$$\cot\theta^L=\cot\theta^R=\begin{cases}\frac{h_y}{h_x} & \text{for nodes 2 and 6}\\ \frac{h_x}{h_y} & \text{for nodes 4 and 8}\end{cases} \tag{A6}$$

Moreover, it is easy to show that the coefficient of U_0 is also independent of the splitting. Hence, the discretization equation (A4) can be written, for arbitrary splittings, as

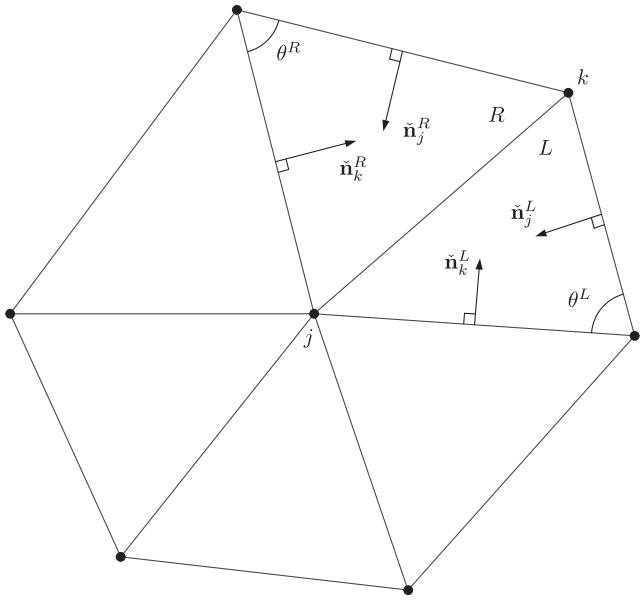


Fig. A1 NC stencil on triangular grids. Note that the normals are not scaled.

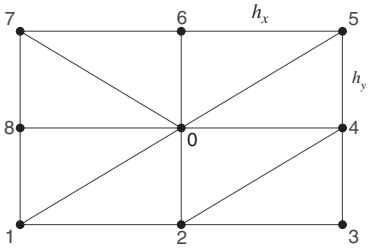


Fig. A2 Triangular grid with arbitrary aspect ratio.

$$\int_\Omega \Delta U \,\mathrm{d}\Omega = -2\left(\frac{h_x}{h_y}+\frac{h_y}{h_x}\right)U_0+\frac{h_y}{h_x}U_2+\frac{h_x}{h_y}U_4+\frac{h_y}{h_x}U_6+\frac{h_x}{h_y}U_8 \tag{A7}$$

$$=h_xh_y\left(\frac{U_4-2U_0+U_8}{h_x^2}+\frac{U_6-2U_0+U_2}{h_y^2}\right) \tag{A8}$$

This is a common five-point finite-difference discretization. Therefore, the NC scheme on grids of types II and III with the arbitrary aspect ratio is equivalent to the common five-point Laplacian. For stretched grids, the corner nodes still do not contribute to the discretization. A similar property holds in 3-D, for which the NC scheme on a tetrahedral grid derived from a (stretched) Cartesian grid by arbitrary diagonal splitting is equivalent to a common seven-point finite-difference discretization.

References

[1] Thomas, J. L., Diskin, B., and Rumsey, C. L., "Towards Verification of Unstructured-Grid Solvers," *AIAA Journal*, Vol. 46, No. 12, Dec. 2008, pp. 3070–3079. doi:10.2514/1.36655

[2] Haselbacher, A. C., "A Grid-Transparent Numerical Method for Compressible Viscous Flow on Mixed Unstructured Meshes," Ph.D. Thesis, Loughborough Univ., Loughborough, Leicestershire, U.K., 1999.

[3] Haselbacher, A. C., McGuirk, J. J., and Page, G. J., "Finite-Volume Discretization Aspects for Viscous Flows on Mixed Unstructured Grids," *AIAA Journal*, Vol. 37, No. 2, 1999, pp. 177–184. doi:10.2514/2.711

[4] Thomas, J. L., Diskin, B., and Nishikawa, H., "A Critical Study of Agglomerated Multigrid Methods for Diffusion on Highly-Stretched Grids," *Computers and Fluids* (accepted for publication).

[5] Holmes, D. G., and Connell, S. D., "Solution of the 2-D Navier–Stokes Equations on Unstructured Adaptive Grids," 9th AIAA CFD Conference, AIAA Paper 89-1392, June 1989.

[6] Rausch, R. D., Batina, J. T., and Yang, H. T., "Spatial Adaptation Procedures on Unstructured Meshes for Accurate Unsteady Aerodynamic Flow Computation," 9th AIAA CFD Conference, AIAA Paper 89-1392, June 1989.

[7] Frink, N. T., "Assesment of an Unstructured-Grid Method For Predicting 3-D Turbulent Viscous Flows," 34th AIAA Aerospace Science Meeting and Exhibit, AIAA Paper 96-0292, Jan. 1996.

[8] Haselbacher, A. C., "On Constrained Reconstruction Operators," 44th AIAA Aerospace Science Meeting and Exhibit, AIAA Paper 2006-1274, Jan. 2006.

[9] Barth, T. J., "Numerical Aspects of Computing High-Reynolds Number Flow on Unstructured Meshes," 29th AIAA Aerospace Science Meeting, AIAA Paper 91-0721, Jan. 1991.

[10] Coirier, W. J., "An Adaptively-Refined, Cartesian, Cell-Based Scheme for the Euler and Navier–Stokes Equations," NASA TM 106754, Oct. 1994.

[11] Frink, N. T., "Tetrahedral Unstructured Navier–Stokes Methods for Turbulent Flows," *AIAA Journal*, Vol. 36, No. 11, 1998, pp. 1975–1982. doi:10.2514/2.324

[12] Mavriplis, D. J., "Revisiting the Least-Square Procedure for Gradient Reconstruction on unstructured Meshes," 18th AIAA CFD Conference, AIAA Paper 2003-3986, June 2003.

[13] Petrovskaya, N. V., "The Choice of Weight Coefficients for Least-Square Gradient Approximation," *Journal of Mathematical Modeling*, Vol. 16, No. 5, 2004, pp. 83–93 (in Russian).

[14] Smith, T. M., Barone, M. F., Bond, R. B., Lorber, A. A., and Baur, D. G., "Comparison of Reconstruction Techniques for Unstructured Mesh Vertex Centered Finite Volume Scheme," 18th AIAA CFD Conference, AIAA Paper 2007-3958, June 2007.

[15] Anderson, W. K., and Bonhaus, D. L., "An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids," *Computers and Fluids*, Vol. 23, No. 1, 1994, pp. 1–21. doi:10.1016/0045-7930(94)90023-X

[16] Diskin, B., and Thomas, J. L., "Accuracy Analysis for Mixed-Element Finite-Volume Discretization Schemes," National Institute of Aerospace TR 2007-08, Hampton, VA, Aug. 2007.

[17] Hirsch, C., *Numerical Computation of Internal and External Flows. Vol. 1, Fundumentals Of Numerical Discretization*, Wiley, New York, 1988.

[18] Syrakos, A., and Goulas, A., "Estimate of the Truncation Error of Finite Volume Discretization of the Navier–Stokes Equations on Collocated Grids," *International Journal for Numerical Methods in Fluids*, Vol. 50, No. 1, 2006, pp. 103–130. doi:10.1002/fld.1038

[19] Turkel, E., "Accuracy of Schemes with Nonuniform Meshes for Compressible Fluid Flows," *Applied Numerical Mathematics*, Vol. 2, No. 6, 1986, pp. 529–550. doi:10.1016/0168-9274(86)90006-1

[20] Giles, M. B., "Accuracy of Node-Based Solutions on Irregular Meshes," *11th International Conference on Numerical Methods in Fluid Dynamics*, edited by D. L. Dwoyer, M. Y. Hussaini, and R. Voigt, Vol. 323, Lecture Notes in Physics, Springer–Verlag, New York, 1989, pp. 369–373.

[21] Forsyth, P. A., and Sammon, P. H., "Quadratic Convergence for Cell-Centered Grids," *Applied Numerical Mathematics*, Vol. 4, No. 5, 1988, pp. 377–394. doi:10.1016/0168-9274(88)90016-5

[22] Kreiss, H. O., Manteuffel, T. A., Wendroff, B., and White, A. B., "Supra-Convergence Schemes on Irregular Grids," *Mathematics of Computations*, Vol. 47, No. 176, Oct. 1986, pp. 537–554.

[23] Manteuffel, T. A., and White, A. B., "The Numerical Solution of the Second-Order Boundary Value Problem on Nonuniform Meshes," *Mathematics of Computations*, Vol. 47, No. 176, Oct. 1986, pp. 511–536.

[24] Diskin, B., and Thomas, J. L., "Accuracy of Gradient Reconstruction on Grids with High Aspect Ratio," National Institute of Aerospace TR 2008-12, Dec. 2008.

[25] Diskin, B., Thomas, J. L., Nielsen, E. J., Nishikawa, H., and White, J. A., "Comparison of Node-Centered and Cell-Centered Unstructured Finite-Volume Discretizations. Part 1: Viscous Fluxes," 47th AIAA Aerospace Science Meeting, AIAA Paper 2009-597, 2009.

Z. J. Wang
Associate Editor

AIAA JOURNAL
Vol. 48, No. 4, April 2010

Critical Study of Agglomerated Multigrid Methods for Diffusion

Hiroaki Nishikawa* and Boris Diskin†
National Institute of Aerospace, Hampton, Virginia 23666
and
James L. Thomas‡
NASA Langley Research Center, Hampton, Virginia 23681

DOI: 10.2514/1.J050055

Agglomerated multigrid techniques used in unstructured-grid methods are studied critically for a model problem representative of laminar diffusion in the incompressible limit. The studied target-grid discretizations and discretizations used on agglomerated grids are typical of current node-centered formulations. Agglomerated multigrid convergence rates are presented using a range of two- and three-dimensional randomly perturbed unstructured grids for simple geometries with isotropic and stretched grids. Two agglomeration techniques are used within an overall topology-preserving agglomeration framework. The results show that a multigrid with an inconsistent coarse-grid scheme using only the edge derivatives (also referred to in the literature as a thin-layer formulation) provides considerable speedup over single-grid methods, but its convergence can deteriorate on highly skewed grids. A multigrid with a Galerkin coarse-grid discretization using piecewise-constant prolongation and a heuristic correction factor is slower and also can be grid dependent. In contrast, nearly grid-independent convergence rates are demonstrated for a multigrid with consistent coarse-grid discretizations. Convergence rates of multigrid cycles are verified with quantitative analysis methods in which parts of the two-grid cycle are replaced by their idealized counterparts.

I. Introduction

MULTIGRID techniques [1] are used to accelerate convergence of current Reynolds averaged Navier–Stokes solvers for steady and unsteady flow solutions, especially for structured-grid applications. Mavriplis [2–4] and Mavriplis and Pirzadeh [5] pioneered agglomerated multigrid methods for large-scale unstructured-grid applications. Impressive improvements in efficiency over single-grid computations have been demonstrated. During a recent development of multigrid methods for unstructured grids [6], it was realized that some of the current approaches for coarse-grid discretization of viscous fluxes used in state-of-the-art codes have serious limitations on highly refined grids. The purpose of this paper is to critically study the current techniques for a simple Poisson equation (representing laminar diffusion in the incompressible limit), assess their performance in grid refinement, and develop improved approaches.

The paper is organized as follows. The model diffusion equation and control-volume partitions are presented from a general finite volume discretization (FVD) standpoint in Sec. II. Elements of multigrid algorithms are described, including a tabulation of target and coarse-grid discretizations in Sec. III. Quantitative analysis methods, in which parts of the actual multigrid cycle are replaced by their idealized counterparts, are described in Sec. IV. The target grids and typical agglomerated grids developed within a topology-preserving framework are shown in Sec. V, followed by two- and three-dimensional results in Secs. VI and VII, respectively. Results from applying analysis methods to 3-D computations are also reported in Sec. VII. Section VIII contains conclusions.

Received 10 July 2009; revision received 18 September 2009; accepted for publication 1 October 2009. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0001-1452/10 and $10.00 in correspondence with the CCC.

*Research Scientist, 100 Exploration Way; hiro@nianet.org.
†100 Exploration Way; bdiskin@nianet.org; currently Visiting Associate Professor, Mechanical and Aerospace Engineering Department, University of Virginia, Charlottesville. Associate Fellow AIAA.
‡Senior Research Scientist, Computational Aerosciences Branch, Mail Stop 128; James.L.Thomas@nasa.gov. Fellow AIAA.

II. Model Diffusion Equation and Boundary Conditions

The FVD schemes considered are derived from the integral form of the diffusion equation,

$$\oint_\Gamma (\nabla U \cdot \hat{\mathbf{n}})\, \mathrm{d}\Gamma = \iint_\Omega f\, \mathrm{d}\Omega \tag{1}$$

where f is a forcing function independent of the solution U, Ω is a control volume with boundary Γ, $\hat{\mathbf{n}}$ is the outward unit normal vector, and ∇U is the solution gradient vector. The boundary conditions are taken as Dirichlet, that is, specified from a known exact solution over the computational boundary. Tests are performed for simple manufactured solutions, namely, collections of polynomial or sine functions. The corresponding forcing functions are found by substituting these solutions into the differential form of the diffusion equation,

$$\Delta U = f \tag{2}$$

and boundary conditions. The *discretization error*, $E_d = U - U^h$, is defined as the difference between the exact continuous solution, U, to the differential Eq. (2) and the exact discrete solution, U^h, of the discretized Eq. (1). The *algebraic error* is the difference between the approximate and exact discrete solutions. A scheme is considered as design-order accurate if its discretization errors computed on a sequence of consistently refined grids [7,8] converge with the design order in the norm of interest.

The general FVD approach requires partitioning the domain into a set of nonoverlapping control volumes and numerically implementing Eq. (1) over each control volume. Node-centered schemes define solution values at the mesh nodes. In two dimensions, the primal meshes are composed of triangular and quadrilateral cells; in three dimensions, the primal cells are tetrahedral, prismatic, pyramidal, or hexahedral. The *median-dual* partition [9,10] used to generate control volumes is illustrated in Fig. 1 for two dimensions. These nonoverlapping control volumes cover the entire computational domain and compose a mesh that is dual to the primal mesh.

The control volumes of each agglomerated grid are found by summing control volumes of a finer grid. Any agglomerated grid can be defined in terms of a conservative agglomeration operator, R_0, as

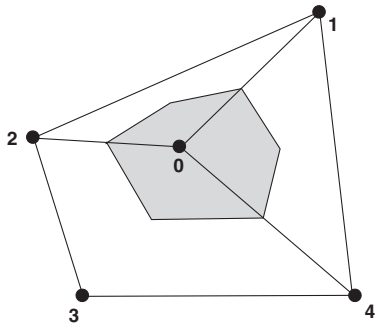


Fig. 1 Illustration of a node-centered median-dual control volume (shaded). Dual faces connect edge midpoints with primal cell centroids. Numbers 0–4 denote grid nodes.

$$\Omega^c = R_0 \Omega^f \tag{3}$$

where the superscripts c and f denote entities on coarser and finer grids, respectively. On the agglomerated grids, the control volumes become geometrically more complex than their primal counterparts and the details of the control-volume boundaries are not retained. The directed area of a coarse-grid face separating two agglomerated control volumes, if required, is found by lumping the directed areas of the corresponding finer-grid faces and is assigned to the virtual edge connecting the centers of the agglomerated control volumes.

III. Multigrid

Elements of the multigrid algorithm are presented in this section. A V cycle [1], denoted as $V(\nu_1, \nu_2)$, uses ν_1 relaxations performed at each grid before proceeding to the coarser grid and ν_2 relaxations after coarse-grid correction; the coarsest grid is solved exactly (with many relaxations). Residuals, r, corresponding to the integral equation (1) are restricted to the coarse grid using R_0, as

$$r^c = R_0 r^f \tag{4}$$

The prolongations P_0 and P_1 are exact for piecewise-constant and linear functions, respectively. The prolongation P_0 is the transpose of R_0. The operator P_1 is constructed locally using linear interpolation from a triangle (two dimensions) or tetrahedra (three dimensions) defined on the coarse grid. The geometrical shape is anchored at the coarser-grid location of the agglomerate that contains the given finer control volume. Other nearby points are found using the adjacency graph. An enclosing simplex is sought that avoids prolongation with nonconvex weights and, in situations in which multiple geometrical shapes are found, the first one encountered is used. Where no enclosing simplex is found, the simplex with minimal nonconvex weights is used. The coarse-grid solution approximation is restricted as

$$U^c = \frac{R_0(U^f \Omega^f)}{\Omega^c} \tag{5}$$

The correction δU to the finer grid is prolonged typically through P_1 as

$$(\delta U)^f = P_1 (\delta U)^c \tag{6}$$

The available consistent target-grid discretizations are the Green–Gauss and the average least squares (Avg-LSQ). These schemes are representative of viscous discretizations used in Reynolds averaged Navier–Stokes unstructured-grid codes. The main target discretization of interest is the Green–Gauss scheme [6], which is the most widely used viscous discretization for node-centered schemes and is equivalent to a Galerkin finite element discretization for triangular/tetrahedral grids. For mixed elements, edge derivatives are used to increase the h ellipticity [1] of the operator and thus avoid checkerboard instabilities [6,10]. Typically, the flux at a face is formed by the edge derivative computed as the divided difference of the solutions at the edge nodes and the Green–Gauss gradient projected onto the directions normal to the edge. The Avg-LSQ scheme defines the flux by the edge derivative and the average of the dual-volume least-squares (LSQ) gradients projected onto the directions normal to the edge [10,11]. The stencils for the dual-volume LSQ gradients include all edge-connected neighbors. The LSQ minimization enforces the given solution at the central node. In both formulations, Dirichlet boundary conditions are implemented strongly.

The exact linear operator is used in the iterative phase of the Green–Gauss scheme, enabling a robust multicolor Gauss–Seidel relaxation. The Avg-LSQ scheme has a comparatively larger stencil, and its exact linearization is not used in iterations; instead, relaxation of the Avg-LSQ scheme relies on an approximate edge-terms-only linearization, which approximates face gradients as edge derivatives. So far, we observe good smoothing rates with this approach, but previous analysis has shown that the smoothing rate can deteriorate on highly skewed grids [6]. The estimates for the smoothing rates obtained with quantitative analysis methods [12] are shown in Sec. VI. The Green–Gauss scheme relies on an element-based data structure and is not considered for agglomerated grids. Note that the Green–Gauss scheme can be written as an edge-based formulation for simplicial grids.

The available coarse-grid discretizations are two possible direct discretizations (Avg-LSQ and edge terms only) and two possible Galerkin discretizations ($R_0 A^f P_0^*$ and $R_0 A^f P_1$) in which the coarse-grid operators are derived from the fine-grid operator. Dirichlet boundary conditions are enforced strongly. The coarse-grid operator is overwritten with the boundary condition linearization at boundary nodes.

The edge-terms-only discretization is often cited as a thin-layer discretization in the literature [2,3,5]; it is a positive scheme but on nonorthogonal grids it is not consistent (i.e., its discrete solution does not converge to the exact continuous solution with consistent grid refinement) [7,8,13]. An orthogonal grid would have each edge node across a face be colinear with the corresponding directed area vector. Another possible coarse-grid discretization strategy, not considered here, is to construct simplicial grids from the coarse-grid vertices.

The Galerkin coarse-grid operator [1] is denoted by RAP. Because the governing equation is a second-order equation, the Galerkin construction, $R_0 A^f P_0$, is formally inconsistent [2,3]; the heuristic correction factor adopted by Mavriplis [2] is used:

$$A^c = R_0 A^f P_0^* = \frac{1}{2} R_0 A^f P_0 \tag{7}$$

The correction factor, applied per agglomerated cell, is derived by enforcing consistency on uniformly agglomerated hexahedral meshes. The Galerkin construction, $R_0 A^f P_1$, is consistent, but was found to be unstable in a multigrid.

IV. Quantitative Analysis of Unstructured Multigrid Solvers

The quantitative analysis methods for unstructured multigrid solvers considered in this section are idealized relaxation (IR) and idealized coarse-grid (ICG) iterations, introduced in [12]. The methods analyze the main complementary parts of a multigrid cycle: relaxation and coarse-grid correction. In a multigrid, relaxation and coarse-grid correction are assigned certain tasks: relaxation is required to smooth the algebraic error, and coarse-grid correction is required to reduce smooth algebraic errors.

To apply the analysis, we first choose a desired sample fine-grid solution (zero is a natural choice for linear problems) and substitute it into the equations to generate the corresponding source and boundary data. Then we form an initial guess (for example, a random perturbation of the solution); thus, the fine-grid algebraic error is known. In the analysis, idealized iterations probe the actual two-grid cycle to identify parts limiting the overall efficiency. In these iterations, one part of the cycle is actual, and its complementary part is replaced with an idealized part. The idealized parts do not depend on the operators to be solved. They are numerical procedures acting

directly on the known algebraic error to efficiently fulfill the task assigned to the corresponding part of the two-grid cycle. The results of the analysis are not single-number estimates; they are rather convergence patterns of the iterations that may either confirm or refute our expectations as to what part of the actual cycle is not efficient in carrying out the assigned task. These IR and ICG analysis methods can be regarded as a numerical extension of the Fourier analysis to problems in which the classical Fourier analysis is inapplicable, in particular, to unstructured-grid solvers.

IR and ICG iterations are analysis methods that test computational efficiency of a two-grid cycle. The two-grid cycle amplification matrix, M, transforms the initial fine-grid algebraic error, $\mathbf{e}^{\text{old}}$, into the after-cycle error, $\mathbf{e}^{\text{new}}$:

$$\mathbf{e}^{\text{new}} = M\mathbf{e}^{\text{old}} \tag{8}$$

The amplification matrix can be defined as

$$M = S^{\nu_2} C S^{\nu_1} \tag{9}$$

Here, ν_1 and ν_2 are small nonnegative integers representing the number of pre- and postrelaxation sweeps, S is the fine-grid relaxation amplification matrix, and C is the amplification matrix of the coarse-grid correction:

$$C = E - P_0 (A^c)^{-1} R_0 A^f \tag{10}$$

where A^c and A^f are the coarse and fine-grid operator matrices, P_0 and R_0 are the prolongation and agglomeration matrices, and E is the fine-grid identity matrix.

For IR iterations, the coarse-grid correction part is actual and the relaxation is idealized. The idealized relaxation may be defined as an explicit error-averaging procedure. In this paper, we employ the IR procedure that replaces the algebraic error at each dual cell with an average of algebraic errors at edge-adjacent cells. At each relaxation step, the known exact solution, if not zero, is subtracted from the current approximation to obtain the algebraic error function. The explicit averaging procedure is applied directly to the error function. The number of sweeps throughout the grid is taken as ν_1 or ν_2, and we denote the corresponding cycles as IR(ν_1, ν_2). The exact solution is then added back. Slow convergence of IR iterations indicates insufficient coarse-grid correction.

In ICG iterations, the relaxation scheme is actual and the coarse-grid correction is idealized. Assuming that the agglomeration and prolongation operators are suitable for efficient multigrid solution, the idealized coarse-grid correction involves idealized fine and coarse operators, A^f_{ideal} and A^c_{ideal}, such that $D^c_\Omega (A^c_{\text{ideal}})^{-1}$ is an accurate approximation to $D^f_\Omega (A^f_{\text{ideal}})^{-1}$ for smooth error components. Here, D^c_Ω and D^f_Ω are diagonal matrices with corresponding coarse- and fine-grid volumes on the diagonals. The simplest idealized operators are corresponding fine- and coarse-grid identity matrices. With this choice, the idealized coarse-grid correction becomes

$$C_{\text{ideal}} = E - P_0 (D^c_\Omega)^{-1} R_0 D^f_\Omega \tag{11}$$

Note that the operator $(D^c_\Omega)^{-1} R_0 D^f_\Omega$ represents volume-weighted averaging. In ICG analysis, the idealized C_{ideal} is applied directly to the known algebraic errors obtained after prerelaxation sweep(s) of the actual relaxation. In implementation, the algebraic error is averaged to the coarse grid, changed in sign, and then prolonged to the fine grid. The slow convergence observed in the ICG iterations is a sign of poor smoothing in relaxation. We denote the ICG cycle as ICG(ν_1, ν_2).

V. Target Grids and Agglomerations

The grids considered are generated by splitting isotropic mapped Cartesian grids into triangular (two-dimensional) or tetrahedral (three-dimensional) elements and then randomly perturbing the grid points by up to one-quarter in two dimensions and one-sixth in three dimensions of the local mesh size. A typical target grid is shown in Fig. 2 for two dimensions with 33 points in each direction. An orthographic view of the boundary grids of a typical target 3-D grid is shown in Fig. 3, again for 33 points in each direction.

The grids are agglomerated within a topology-preserving framework, in which hierarchies are assigned based on connections to the computational boundaries. Corners are identified as grid points with three or more boundary-condition-type closures (or three or more boundary slope discontinuities). Ridges are identified as grid points with two boundary-condition-type closures (or two boundary slope discontinuities). Valleys are identified as grid points with a single boundary-condition-type closure, and interiors are identified as grid points with no boundary closure. The agglomerations proceed hierarchically from seeds within the topologies, first corners, then ridges, then valleys, and finally interiors. Rules are enforced to maintain the boundary condition types of the finer grid within the agglomerated grid. Candidate volumes to be agglomerated are vetted against the hierarchy of the currently agglomerated volumes using the rules summarized in Table 1. The allowed entries denote that interior volumes can be agglomerated to any existing agglomerate. The single disallowed entry enforces that two corners cannot be agglomerated. The conditional entries denote that further inspection of the connectivity of the topology must be considered before agglomeration is allowed. For example, a ridge can be agglomerated into a corner if the ridge is part of the boundary condition

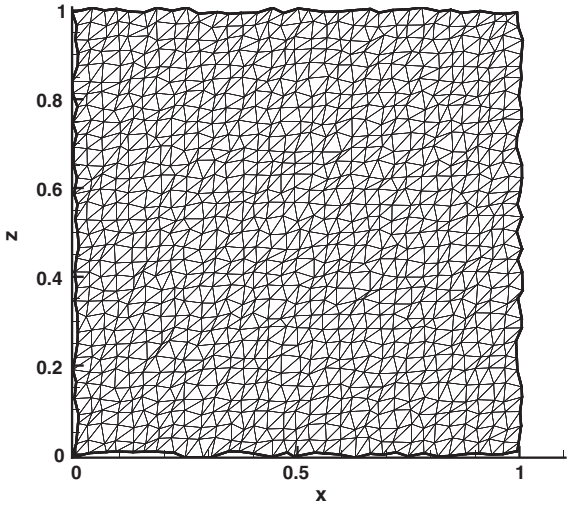


Fig. 2 Typical 2-D target grid.

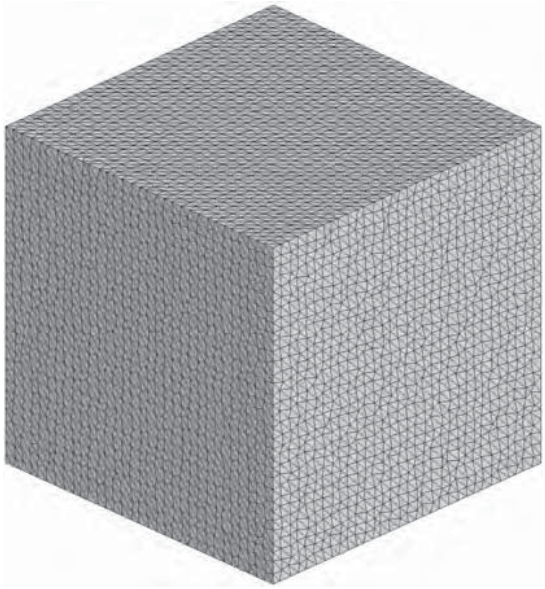


Fig. 3 Orthographic view of a typical 3-D target grid.

Table 1 Admissible agglomerations

Hierarchy of agglomeration	Hierarchy of added volume	Agglomeration decision
Corner	Interior	Allowed (corner to interior)
Corner	Valley	Conditional
Corner	Ridge	Conditional
Corner	Corner	Disallowed (two corners)
Ridge	Interior	Allowed (ridge to interior)
Ridge	Valley	Conditional
Ridge	Ridge	Conditional
Valley	Interior	Allowed (valley to interior)
Valley	Valley	Conditional
Interior	Interior	Allowed (interior to interior)

specification associated with the corner. As another example, a ridge can be agglomerated into an existing ridge agglomeration if the two boundary conditions associated with each ridge are the same. Also, the prolongation operator P_1 is modified to prolong only from hierarchies equal to or above the hierarchy of the prolonged point. Hierarchies on each agglomerated grid are inherited from the finer grid.

There are two agglomeration schemes, referred to as schemes I and II, that have evolved historically within this development. The agglomeration scheme I orders the possible points within a hierarchy using the distance from the corners of the grid and the closest points are taken first. Given a seed, a triad is constructed using a surrounding cloud of points, defined from the adjacency list. The first leg of the triad is defined by the seed and the nearest point. The next leg of the triad is defined by including another point from the entries in the cloud such that the leg is most orthogonal to the first leg. The third leg is found as the one most parallel to the cross product of the first two legs. Points within the volume defined by the triads (extended to infinite length) are taken, first for the edge adjacencies in the cloud and subsequently for the entire adjacency, to satisfy a global coarsening goal (four volumes agglomerated for two dimensions and eight for three dimensions). The agglomeration scheme II also starts from the corners. After all corners have been agglomerated, a front list is defined by collecting nodes adjacent to the agglomerated corners. It then proceeds to agglomerate nodes in the list (while updating the list as the agglomeration proceeds) in the following order: ridges, valleys, interiors. A node is selected among those in the same hierarchy that has the least number of nonagglomerated neighbors to reduce the occurrences of agglomerations with small numbers of volumes. For a given seed, it collects all neighbors and agglomerates them up to a specified maximum number, for example, eight in three dimensions. The agglomeration continues until the front list becomes empty. For either agglomeration scheme, agglomerations containing only a few volumes are combined with other agglomerations, as is typical of the methods used in the literature.

Figure 4 shows three agglomerated grids generated from the primal grid in Fig. 2 using agglomeration schemes I and II. Figure 5 shows three agglomerated grids generated from the primal grid in Fig. 3 using agglomeration scheme II. The agglomerations are representative of those in the literature.

For meshes stretched toward a surface, implicit lines are used. They are defined in the direction normal to the surface by the shortest distance between nodes, constructed on the primal grid, and terminated in the isotopic region [1–3]. The agglomerations are first constructed along the boundary of the grid (corners, ridges, and valleys) and then the cells are agglomerated from the boundary within the implicit lines associated with the stretched grid. The

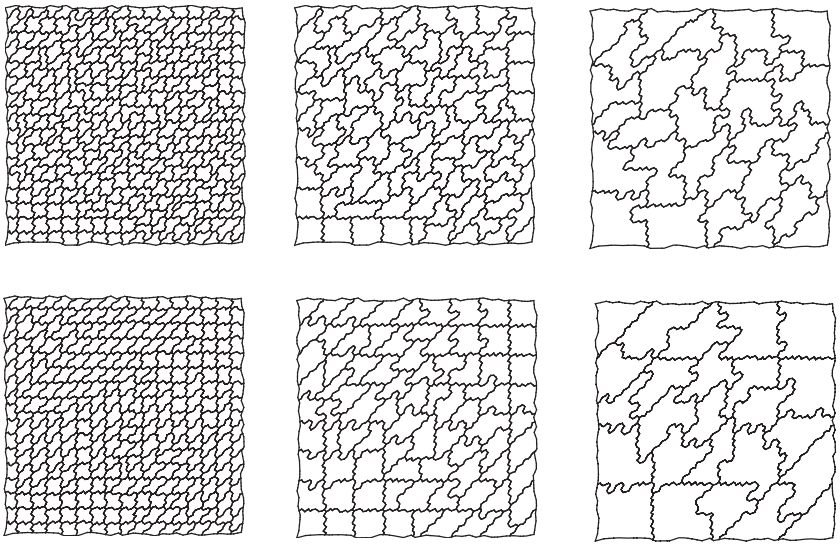


Fig. 4 Control-volume boundaries (nonlumped) for 2-D agglomerations using scheme I (top row) and scheme II (bottom row).

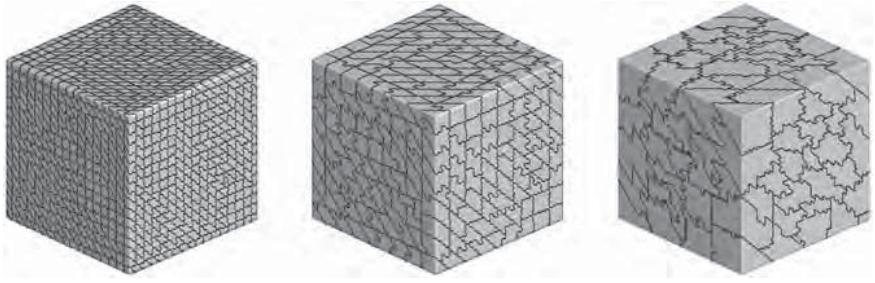


Fig. 5 Control-volume boundaries (nonlumped) for 3-D agglomerations using scheme II.

Table 2 Summary of multilevel asymptotic convergence rates per $V(2,1)$ multigrid cycle with agglomeration scheme I for the Green–Gauss scheme on the fine grid with various coarse-grid operators; cycles to convergence are in parentheses

Fine grid	Direct discretization		Galerkin discretization	
	Avg-LSQ	Edge terms only	$R_0A^fP_0^*$	$R_0A^fP_1$
33×33	0.15(12)	0.20(13)	0.51(23)	Divergent
65×65	0.18(12)	0.29(15)	0.58(25)	Divergent
129×129	0.21(12)	0.33(16)	0.60(24)	Divergent
257×257	0.19(12)	0.44(18)	0.62(24)	Divergent

Table 3 Summary of multilevel asymptotic convergence rates per $V(2,1)$ multigrid cycle with agglomeration scheme II for the Green–Gauss scheme on the fine grid with various coarse-grid operators; cycles to convergence are in parentheses

Fine grid	Direct discretization		Galerkin discretization	
	Avg-LSQ	Edge terms only	$R_0A^fP_0^*$	$R_0A^fP_1$
33×33	0.16(11)	0.29(15)	0.47(23)	Divergent
65×65	0.16(11)	0.42(19)	0.58(27)	Divergent
129×129	0.18(12)	0.54(26)	0.68(31)	Divergent
257×257	0.18(12)	0.82(60)	0.71(34)	Divergent

Table 4 Summary of asymptotic convergence rates per relaxation and the number of relaxations to converge in single-grid calculations (the Green–Gauss scheme is used)

	Convergence per relaxation	Number of relaxations
33×33	0.99710	1278
65×65	0.99926	5440
129×129	0.99945	16320

boundary agglomerate is merged with the volumes corresponding to the next node in the line. The agglomeration continues to the end of the shortest line in the boundary agglomerate, merging two cells in the normal direction at a time. After agglomeration of lines, the algorithm uses the point agglomeration method for the rest of the domain. Illustrations of stretched grids and corresponding agglomerations are shown in Section VI.

VI. Two-Dimensional Results

A summary of $V(2,1)$ multigrid cycle convergence rates is compiled in Tables 2 and 3 for the two agglomeration schemes, respectively. The computations are performed for the Green–Gauss scheme on the fine grid with various coarse-grid operators. The asymptotic convergence per cycle and the number of cycles to reach machine-precision residuals from a random initial perturbation are tabulated. Multigrid cycles employ as many levels as possible; for example, there are six levels used for the 129×129 target grid and four levels for the 33×33 target grid. Table 4 shows convergence rates per relaxation and the number of relaxations to converge for single-grid calculations. Somewhat surprisingly, with the Galerkin coarse-grid operator constructed via $R_0A^fP_1$, the multigrid algorithm is divergent. The reason, confirmed by analysis, is that the coarse-grid operator, although accurate, loses h ellipticity [1]. This loss of h ellipticity for the Galerkin operator with simplex-based P_1 prolongation has been observed even with quadrilateral grids, for which bilinear prolongation is known to result in h elliptic coarse-grid operators.

With the Galerkin coarse-grid operator $R_0A^fP_0^*$, the multigrid algorithm is stable. However, the convergence rates degrade on finer grids with either agglomeration scheme. With the coarse-grid operator using only the edge terms, the convergence per cycle is generally better, but again shows a deterioration on finer grids. The deterioration is noticeably worse with the agglomeration scheme II, although it is hard to judge the reason from visual inspection of the agglomerated grids. With the Avg-LSQ scheme, the convergence per cycle is 0.21 or better and grid independent. In any case, the multigrid algorithm, whether grid dependent or grid independent, gives considerable speedup over a single-grid method; compare Tables 2 and 3 with Table 4.

The dependence on the number of levels in the multigrid cycle is shown in Table 5 using the two agglomeration schemes. In all cases, the coarsest-grid residual was reduced 2 orders of magnitude from the initial coarsest-grid residual; the results were insensitive to reducing the coarsest-level residual further. Typically, convergence in a two-level cycle is a lower bound of the convergence in a multilevel cycle; such behavior is observed with the coarse grids discretized using the Avg-LSQ scheme. The observed multilevel cycle convergence is very similar to the two-level cycle convergence. With the coarse grids discretized using the edge-terms-only scheme, the results are unexpected; the six-level cycle convergence is significantly better than the two-level cycle convergence. This is true for both agglomeration schemes, although the effect is considerably more pronounced with agglomeration scheme II. A possible explanation is that the coarser agglomeration grids have a less consistently high skewing, thus mitigating inconsistency of the edge-terms-only discretization. Although we did not tabulate the results, the dependence on the number of levels in the multigrid cycle for the heuristic Galerkin construction is more or less as would be expected; the two-level cycle converges best, and performance falls off with increasing number of levels.

The grid-dependent convergence of multigrid cycles with the edge-terms-only scheme (Tables 2 and 3) is attributed to the poor coarse-grid correction, which is confirmed by quantitative analysis. Both ICG and IR were applied to a family of element-based grids (33×33, 65×65, 129×129, and 257×257) with coarser grids constructed in turn using each of the two agglomeration schemes. Convergence of the ICG(3,3) scheme was less than 0.1 per cycle in all cases, indicating that the multicolor relaxation is not a source of the grid-dependent convergence. The results of applying IR(3,3) are shown in Table 6 with the coarse-grid correction using the Avg-LSQ and the edge-terms-only schemes for each of the two agglomeration schemes. With the coarse-grid correction using the Avg-LSQ scheme, the convergence rates per cycle are grid independent and

Table 5 Asymptotic convergence per $V(2,1)$ cycle for the Green–Gauss scheme on the target 129 × 129 grid with various coarse-grid operators; cycles to convergence are in parentheses

	Agglomeration scheme I		Agglomeration scheme ii	
	Coarse-grid discretization		Coarse-grid discretization	
Multigrid levels	Avg-LSQ	Edge terms only	Avg-LSQ	Edge terms only
6	0.21(12)	0.33(16)	0.18(12)	0.54(26)
5	0.21(12)	0.33(16)	0.18(12)	0.54(26)
4	0.20(12)	0.35(16)	0.18(12)	0.60(30)
3	0.19(12)	0.43(19)	0.18(12)	0.69(39)
2	0.18(12)	0.41(18)	0.17(12)	0.81(55)

Table 6 Asymptotic convergence per cycle using IR(3,3) analysis; cycles to convergence are in parentheses

	Agglomeration scheme I		Agglomeration scheme ii	
	Coarse-grid discretization		Coarse-grid discretization	
Element-based grid	Avg-LSQ	Edge terms only	Avg-LSQ	Edge terms only
33×33	0.11 (9)	0.32 (15)	0.14 (10)	0.55 (26)
65×65	0.13 (10)	0.49 (21)	0.15 (10)	0.72 (44)
129×129	0.20 (11)	0.54 (26)	0.21 (12)	>0.99 (>200)
257×257	0.17(10)	0.61 (28)	0.20 (11)	>0.99 (>500)

Table 7 Asymptotic convergence per cycle using ICG (3,3) analysis for family of agglomerated grids; cycles to convergence are in parentheses (the Avg-LSQ scheme is used on all grids)

	Agglomeration scheme	
Agglomeration level	I	II
4	0.06 (8)	0.06 (8)
3	0.06 (8)	0.05 (7)
2	0.07 (8)	0.07 (8)
1	0.06 (8)	0.08 (8)
0	0.07 (8)	0.08 (8)

Table 8 Asymptotic convergence for two-level cycle for sheared primal grids; cycles to convergence are in parentheses (the Green–Gauss scheme is used on the primal grids)

	Agglomeration scheme I		Agglomeration scheme II	
	Coarse-grid discretization		Coarse-grid discretization	
Primal grid	Avg-LSQ	Edge terms only	Avg-LSQ	Edge terms only
17×17	0.17(12)	0.48(27)	0.15(12)	0.63(39)
33×33	0.18(12)	0.67(40)	0.20(12)	unstable
65×65	0.21(12)	0.88(102)	0.22(13)	unstable

better than 0.21; the number of cycles to convergence is 12 at most. With the coarse-grid correction using the edge-terms-only scheme, the convergence rates and number of cycles to converge are grid dependent.

With a consistent coarse-grid discretization, such as the Avg-LSQ scheme, we expect good two-level convergence rates. With the Avg-LSQ scheme, relaxation is implemented within a defect-correction setting in which the approximate linearization based on the edge-terms-only scheme is used as a driver. The viability of this approach is checked using ICG(3,3) for the family of grids agglomerated from the parent 257×257 grid. The convergence per cycle is shown in Table 7 for different agglomeration levels, where the target element-based grid is denoted as level 0. In all cases, the edge-terms-only scheme provides adequate relaxation, yielding an order of magnitude convergence per ICG(3,3) cycle.

The spatial convergence of discretization error for agglomerate families with the Avg-LSQ target-grid discretization is shown in Fig. 6. Results with the edge-terms-only discretization are also shown for reference. The manufactured solution is $U = \sin(\pi x + 0.8\pi y) + 0.1x + 0.2y$ and the coarser grids were generated using agglomeration scheme II. Each agglomerate family is composed of a target element-based grid and agglomerated grids generated recursively; a particular agglomerate family is denoted by the density of the primal mesh in parentheses. The L_1 norm of the discretization error is shown versus an equivalent mesh size, taken as the L_1 norm of a local characteristic distance, that is, $h_V = \|\Omega^{1/d}\|$, where d is the number of spatial dimensions. The edge-terms-only discretization shows no order property, as expected, but the Avg-LSQ scheme shows a second-order convergence of discretization errors. Thus, the Avg-LSQ scheme is second-order accurate and provides a viable way of discretizing diffusion terms on agglomerated coarse grids.

For the finer agglomerate family, multigrid convergence is shown in Fig. 7 using the Avg-LSQ discretization on all grids. Multilevel $V(2, 2)$ cycles are used with two levels on the coarsest agglomerate and six levels on the primal mesh. The initial conditions are taken as the exact solution with a randomly perturbed error on each grid. Grid-independent convergence is shown with approximately an order of magnitude reduction in residual per cycle.

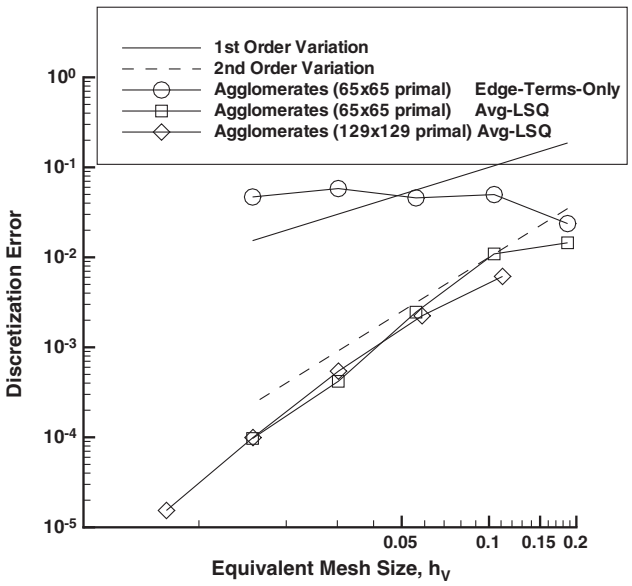


Fig. 6 Spatial convergence of discretization error for agglomerate families with the Avg-LSQ and edge-terms-only schemes.

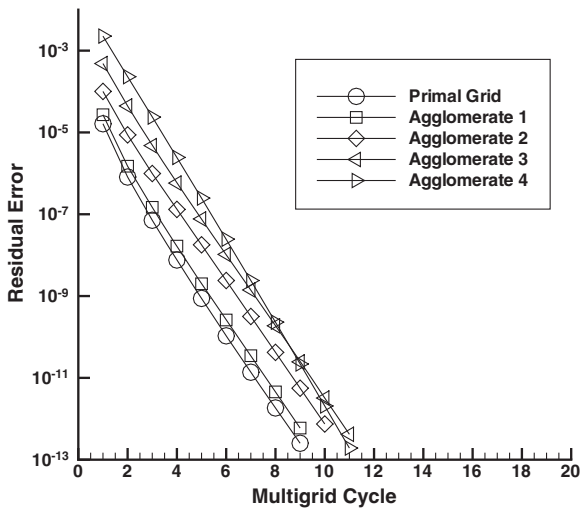
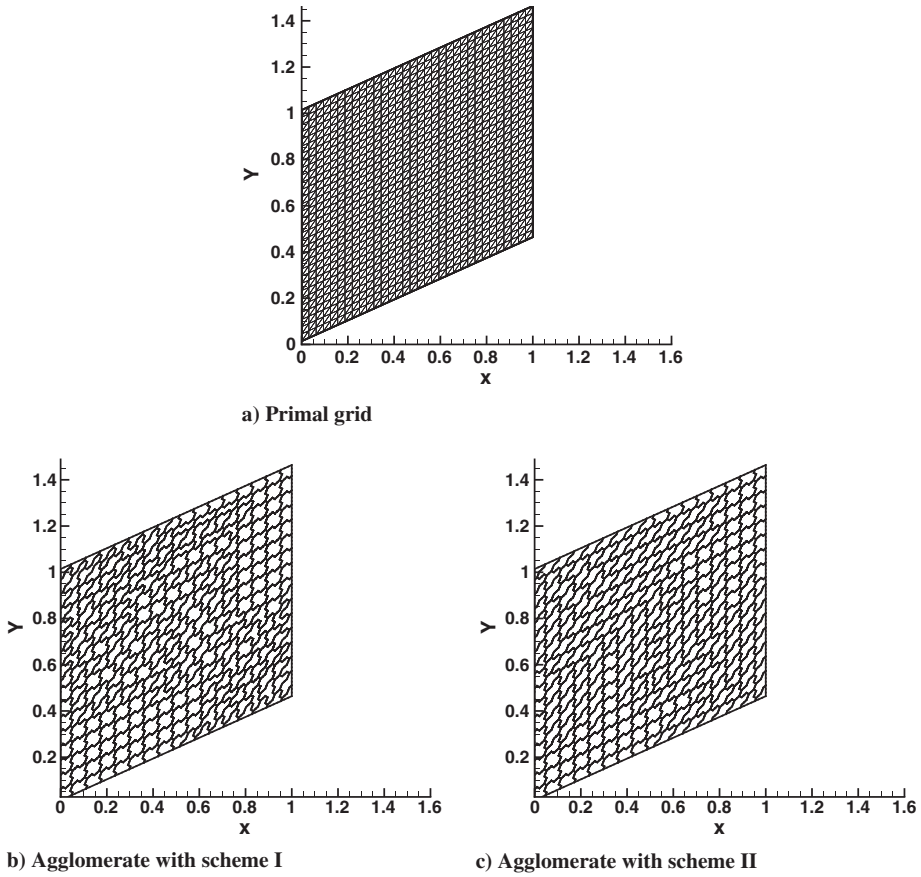


Fig. 7 Multigrid convergence for agglomerate family composed of the 129 × 129 primal mesh and its coarsened agglomerates, using the Avg-LSQ scheme in all levels.



a) Primal grid

b) Agglomerate with scheme I

c) Agglomerate with scheme II

Fig. 8 Primal (33 × 33) grid and agglomerated grids.

Finally, to demonstrate that multigrid convergence with the coarse-grid edge-terms-only discretization is grid dependent, a series of sheared primal grids is considered with skew angles consistently greater than 45 deg. A typical primal grid and the agglomerated grids using the two agglomeration schemes are shown in Fig. 8. The convergence of two-level multigrid cycles is shown in Table 8 using the two agglomeration schemes with different coarse-grid discretizations. Convergence with the coarse-grid Avg-LSQ discretization is very similar using either agglomeration scheme and nominally grid independent. With the coarse-grid discretized using the edge-terms-only scheme, the convergence is grid-dependent for agglomeration scheme I; the multigrid cycle is unstable beyond the coarsest grid with agglomeration scheme II. Note the variability in convergence with the edge-terms-only coarse-grid discretization between agglomeration schemes I and II even though the agglomerations from the two schemes are quite regular and similar (Fig. 8).

VII. Three-Dimensional Results

Multigrid asymptotic convergence rates are shown in Table 9 with various coarse-grid operators for a range of isotropic 3-D grids ($9 \times 9 \times 9$ to $129 \times 129 \times 129$). Results are obtained with multiple-level $V(3, 3)$ multigrid cycles. Two-grid results are not shown but are very similar to the multiple-level results. Agglomerated grids are generated with scheme II.

Table 9 Summary of multilevel asymptotic convergence rates per $V(3, 3)$ multigrid cycle with agglomeration scheme II for the Green–Gauss scheme on the fine grid with various coarse-grid operators

Fine grid	Direct discretization		Galerkin discretization	
	Avg-LSQ	Edge terms only	$R_0A^fP_0^*$	$R_0A^fP_1$
$9 \times 9 \times 9$	0.05	0.05	0.15	divergent
$17 \times 17 \times 17$	0.11	0.16	0.35	divergent
$33 \times 33 \times 33$	0.14	0.26	0.54	divergent
$65 \times 65 \times 65$	0.16	0.30	0.67	divergent
$97 \times 97 \times 97$	0.24	0.33	0.73	divergent
$129 \times 129 \times 129$	0.22	0.34	0.76	divergent

The 3-D results are consistent with the 2-D results. With the Galerkin coarse-grid operator constructed via $R_0A^fP_0^*$, the multigrid algorithm is stable, but the convergence degrades on finer grids. The Galerkin coarse-grid operator constructed via $R_0A^fP_1$ was again found to be divergent. With agglomerated grids using the edge-terms-only scheme, the convergence per cycle is better but again shows a deterioration on finer grids. Note that the deterioration observed in three dimensions is weaker than that in two dimensions. With agglomerated grids using the Avg-LSQ scheme, the convergence per cycle is practically grid-independent; the asymptotic convergence per cycle is similar to that in two dimensions. In any case, the multigrid method gives considerable speedup over a single-grid method, as clearly seen in Fig. 9, which shows the residual convergence versus work units for the $65 \times 65 \times 65$ grid case. Here, the work unit is defined as the work required for one residual evaluation and relaxation on the target grid; a multigrid $V(3, 3)$ cycle requires about 7 work units; restriction and prolongation work is small and has been neglected. The multigrid method converged in 108 work units using the Avg-LSQ scheme, 144 using the edge-terms-only scheme, and 425 with the Galerkin coarse-grid operator constructed via $R_0A^fP_0^*$, whereas the single-grid method converged in 10,335 work units. Some dependence on the number of levels in the multigrid cycle similar to that for 2-D cases as shown in Table 5 was observed also in three dimensions, but the variation was smaller.

The multigrid $V(3, 3)$ cycle is tested with a line agglomeration/relaxation for stretched grids typical in high-Reynolds-number flow simulations. The grids are regular tetrahedral $9 \times 9 \times 17$, $13 \times 13 \times 25$, $17 \times 17 \times 33$, $24 \times 24 \times 47$, $33 \times 33 \times 65$, $49 \times 49 \times 97$ grids with exponential stretching applied in the z direction. The stretching is applied only in the lower half region; the upper half remains isotropic. A representative grid is shown in Fig. 10. A line

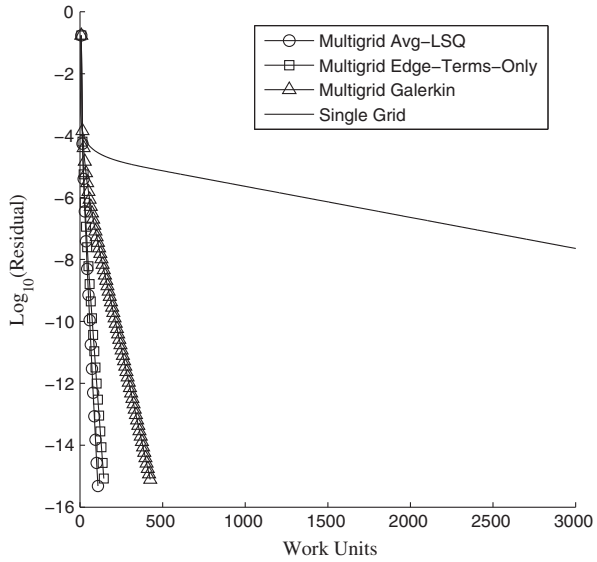


Fig. 9 Residual versus work units for the $V(3,3)$ multilevel multigrid methods with agglomeration scheme II and a single-grid method on the 65 × 65 × 65 grid. The Green–Gauss scheme is used on the fine grid.

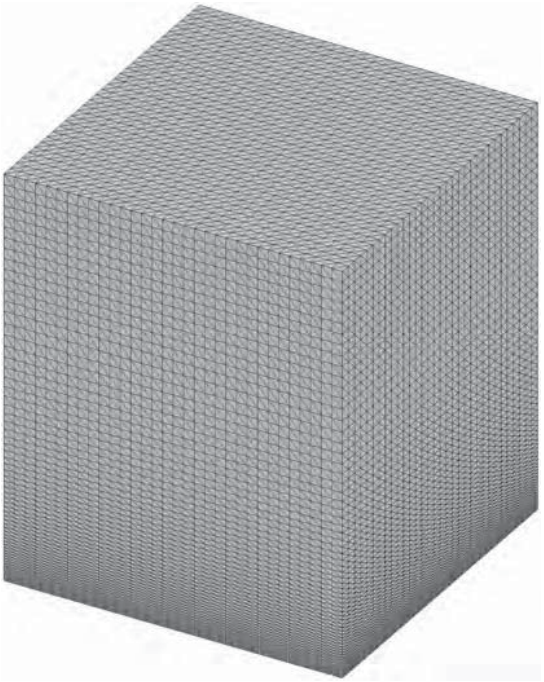


Fig. 10 33 × 33 × 65 stretched grid with the maximum aspect ratio of 6.25.

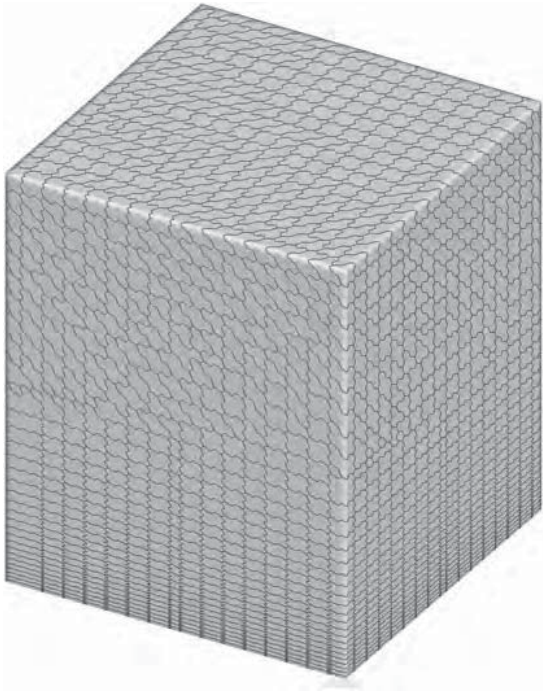


Fig. 11 Coarse grid for the 33 × 33 × 65 stretched grid with the maximum aspect ratio of 6.25.

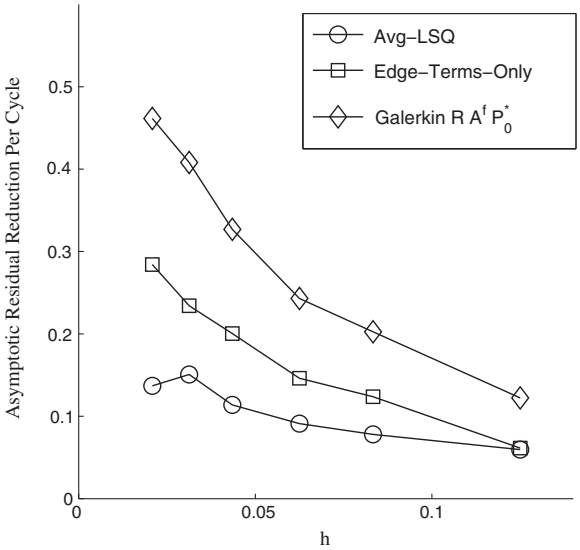


Fig. 12 Asymptotic convergence rate per $V(3,3)$ multigrid two-level cycle with agglomeration scheme II and a line agglomeration/relaxation in the stretched region. The Green–Gauss scheme is used on the fine grid.

agglomeration and a line relaxation are applied in the stretched region. A representative coarse grid is shown in Fig. 11. The results are shown in Fig. 12. The mesh size h corresponds to $1/(N^{1/3}-1)$, where N is the total number of nodes. Again, multigrid with either the edge-terms-only or the Galerkin coarse-grid operator shows a deterioration on finer grids, whereas a multigrid with the Avg-LSQ scheme gives nearly grid-independent results. One would have to consider even higher mesh densities to clearly indicate the behavior of the convergence rate with mesh refinement.

The IR and ICG analysis methods have been applied within a two-grid multigrid cycle on perturbed isotropic tetrahedral grids to evaluate relaxation smoothing and efficiency of coarse-grid correction. The point relaxation scheme has been tested on a 33 × 33 × 33 grid for three formulations: Green–Gauss, Avg-LSQ, and edge terms only. Convergence rates observed in ICG iterations and collected in Table 10 show that the tested relaxation is an efficient error smoother for all three schemes; the high-frequency error reduction is better than 0.55, which is an excellent smoothing factor.

IR iterations have been performed to analyze the quality of coarse-grid correction with two different coarse-grid schemes: Avg-LSQ and edge-terms-only approximation. The results are shown in Table 11. To provide robust grid-independent convergence rates in a

Table 10 Summary of smoothing rates of three relaxation schemes obtained from ICG(1,0) on a 33 × 33 × 33 perturbed isotropic tetrahedral grid (the Green–Gauss scheme is used on the fine grid)

Green–Gauss	Avg-LSQ	Edge-terms-only
0.545	0.470	0.358

Table 11 Summary of convergence rates for two coarse-grid correction schemes obtained from IR(3,3) on a 33 × 33 × 33 perturbed isotropic tetrahedral grid

Coarse grid	Avg-LSQ	Edge terms only
P_0 prolongation	0.124	0.303
P_1 prolongation	0.125	0.303

multigrid cycle, the coarse-grid correction is expected to reduce smooth errors by an order of magnitude. Convergence rates observed in IR iterations with six explicit error-averaging sweeps show that the coarse-grid correction is adequate for the Avg-LSQ scheme. The rates observed for the edge-terms-only scheme are slow and further deteriorate on grids with consistent high skewing. Both schemes appear insensitive to the prolongation order, demonstrating almost identical convergence rates for either P_0 or P_1 prolongation operator.

VIII. Conclusions

Agglomerated multigrid techniques used in unstructured-grid methods have been critically studied for a model problem representative of laminar diffusion in the incompressible limit. The studied target-grid discretizations and discretizations used on agglomerated grids are typical node-centered formulations. Agglomerated multigrid convergence rates are compiled using a range of two- and three-dimensional randomly perturbed unstructured grids for simple geometries, including isotropic and stretched grids. Two agglomeration techniques are used within an overall topology-preserving agglomeration framework. The results show that a multigrid with an inconsistent coarse-grid scheme using only the edge terms (also referred to in the literature as a thin-layer formulation) provides considerable speedup over single-grid methods, but its convergence can deteriorate on consistently skewed grids. A multigrid with a formally inconsistent Galerkin coarse-grid discretization using piecewise-constant prolongation and a heuristic correction is slower and also can be grid dependent. A consistent Galerkin coarse-grid construction using simplex prolongation was found to be unstable because the discretization lacked h ellipticity. Nearly grid-independent convergence rates are demonstrated for a multigrid with consistent coarse-grid discretizations. Additional study with higher mesh densities is required to determine grid-independence for 3-D high-aspect-ratio grids. The results from the actual cycle are verified using discrete analysis methods in which parts of the cycle are replaced by their idealized counterparts.

Acknowledgments

The three-dimensional results presented were computed within the FUN3D suite of codes at NASA Langley Research Center.[§] The contributions of E. J. Nielsen, J. A. White, and R.T. Biedron of NASA to the implementation within FUN3D are gratefully acknowledged. Nishikawa was supported by the National Institute of Aerospace under the NASA Fundamental Aeronautics Program through NASA Research Announcement Contract NNL07AA23C. Diskin was supported by the National Institute of Aerospace under NASA Fundamental Aeronautics Program through NASA Research Announcement Contract NNL07AA31C.

References

[1] Trottenberg, U., Oosterlee, C. W., and Schüler, A., *Multigrid*, Academic Press, London, 2000.

[2] Mavriplis, D. J., "Multigrid Techniques for Unstructured Meshes," VKI Lecture Series, Von Karman Institute for Fluid Dynamics VKI-LS 1995-02, Rhode-Saint-Genese, Belgium 1995.

[3] Mavriplis, D. J., "Unstructured Grid Techniques," *Annual Review of Fluid Mechanics*, Vol. 29, 1997, pp. 473–514. doi:10.1146/annurev.fluid.29.1.473

[4] Mavriplis, D. J., "An Assessment of Linear Versus Non-Linear Multigrid Methods for Unstructured Mesh Solvers," *Journal of Computational Physics*, Vol. 175, 2002, pp. 302–325. doi:10.1006/jcph.2001.6948

[5] Mavriplis, D. J., and Pirzadeh, S., "Large-Scale Parallel Unstructured Mesh Computations for 3D High-Lift Analysis," *Journal of Aircraft*, Vol. 36, No. 6, 1999, pp. 987–998. doi:10.2514/2.2540

[6] Diskin, B., Thomas, J. L., Nishikawa, H., Nielsen, E. N., and White, J. A., "Comparison of Node-Centered and Cell-Centered Unstructured Finite-Volume Discretizations Part I Viscous Fluxes," AIAA Paper 2009-597, Jan. 2009.

[7] Diskin, B., and Thomas, J. L., "Accuracy Analysis for Mixed-Element Finite-Volume Discretization Schemes," National Institute of Aerospace Technical Report 2007-8, Aug. 2007.

[8] Thomas, J. L., Diskin, B., and Rumsey, C. L., "Towards Verification of Unstructured Grid Methods," *AIAA Journal*, Vol. 46, No. 12, December 2008, pp. 3070–3079. doi:10.2514/1.36655; also AIAA Paper 2008-0666.

[9] Barth, T. J., "Numerical Aspects of Computing High-Reynolds Number Flow on Unstructured Meshes," AIAA Paper 91-0721, Jan. 1991.

[10] Haselbacher, A. C., "A Grid-Transparent Numerical Method for Compressible Viscous Flow on Mixed Unstructured Meshes," Ph.D. Thesis, Loughborough Univ., Loughborough, England, UK, 1999.

[11] Weiss, J. M., Maruszeski, J. P., and Smith, W. A., "Implicit Solution of Preconditioned Navier–Stokes Equations Using Algebraic Multigrid," *AIAA Journal*, Vol. 37, No. 1, Jan. 1999, pp. 29–36. doi:10.2514/2.689

[12] Diskin, B., Thomas, J. L., and Mineck, R., "On Quantitative Analysis Methods for Multigrid Solutions," *SIAM Journal on Scientific Computing*, Vol. 27, No. 1, 2005, pp. 108–129. doi:10.1137/030601521

[13] Svärd, M., Gong, J., and Nordström, J., "An Accuracy Evaluation of Unstructured Node-Centered Finite-Volume Methods," *Applied Numerical Mathematics*, Vol. 58, No. 8, 2008, pp. 1142–1158. doi:10.1016/j.apnum.2007.05.002; also available as National Institute of Aerospace Rept. 2005-04, NASA CR-2006-214293, April 2006.

W. Anderson
Associate Editor

[§]Data available online at http://fun3d.larc.nasa.gov/ [retrieved 4 December 2009].

50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition
09 - 12 January 2012, Nashville, Tennessee

AIAA 2012-0609

Effects of mesh regularity on accuracy of finite-volume schemes

Boris Diskin* James L. Thomas†

The effects of mesh regularity on the accuracy of unstructured node-centered finite-volume discretizations are considered. The focus of this paper is on an edge-based approach that uses unweighted least-squares gradient reconstruction with a quadratic fit. Gradient errors and discretization errors for inviscid and viscous fluxes are separately studied according to a previously introduced methodology. The methodology considers three classes of grids: isotropic grids in a rectangular geometry, anisotropic grids typical of adapted grids, and anisotropic grids over a curved surface typical of advancing-layer viscous grids. The meshes within these classes range from regular to extremely irregular including meshes with random perturbation of nodes. The inviscid scheme is nominally third-order accurate on general triangular meshes. The viscous scheme is a nominally second-order accurate discretization that uses an average-least-squares method. The results have been contrasted with previously studied schemes involving other gradient reconstruction methods such as the Green-Gauss method and the unweighted least-squares method with a linear fit. Recommendations are made concerning the inviscid and viscous discretization schemes that are expected to be least sensitive to mesh regularity in applications to turbulent flows for complex geometries.

I. Introduction

Traditional mesh-quality metrics tend to assess meshes without taking into account the type of equations being solved, solutions, or the desired computational output. The most widely-used mesh quality metrics are geometric in nature, considering shape, size, angles, aspect ratio, skewness, Jacobian, etc., of the mesh elements. Additional considerations include variations between mesh elements, such as cell-to-cell and face-to-face ratios and line smoothness, etc. There is a widespread perception that the most accurate and efficient solutions are obtained on "pretty" meshes similar to either structured Cartesian meshes or to meshes composed from identical perfect elements (perfect triangles, tetrahedrals, etc.) This perception contradicts modern Computational Fluid Dynamics (CFD) practice, in which accurate solutions are computed on practical meshes that would be characterized as unacceptable by many geometric mesh quality metrics. Moreover, the most powerful state-of-art method for improving solution accuracy, output-based mesh adaptation,[1] tends to produce "ugly" meshes but provides vast improvements of the accuracy-per-degree-of-freedom ratio.[2] It is widely recognized today that mesh quality indicators should involve information about the solution[3–5] and, more generally, the discretization method in use and the desired computational output.

Historically, mesh quality analyses were first performed for finite-difference and finite-element methods. It is not straightforward to translate those approaches to finite-volume discretizations (FVD) that represent the state of art in CFD computations. While there is no doubt that certain mesh characteristics critically affect accuracy of CFD solutions and gradients, the precise nature of this influence (what affects what) is far from clear.

For finite-difference approaches, most of the mesh quality methods try to establish connections between mesh and truncation error.[6,7] The truncation error analysis is often applied to FVD schemes as well.[8] However, it has been long known, that truncation errors of FVD schemes on unstructured grids are not reliable estimators of discretization

*National Institute of Aerospace (NIA), 100 Exploration Way, Hampton, VA 23681, USA, Associate Fellow AIAA bdiskin@nianet.org. Also Department of Mechanical and Aerospace Engineering, University of Virginia, Charlottesville, VA 22904, USA. Supported by NASA Fundamental Aeronautics Program, Supersonics Project, NRA Contract NNL07AA23C (PI: Prof. N. K. Yamaleev).

†Computational AeroSciences Branch, NASA Langley Research Center, Mail Stop 128, Fellow AIAA, James.L.Thomas@nasa.gov.

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

errors. The *supra-convergence* of discretization errors observed and studied for at least 50 years (e.g., see the list of references in Ref. 8) indicates that design-order accurate FVD solutions can be computed on unstructured grids even when truncation errors exhibit a lower-order convergence or, in some cases, do not converge at all.[9–11]

The theory and applications of mesh quality assessments are well developed and widely used within the finite-element community. While groundbreaking work focused on pure geometrical mesh-quality metrics, such as large angles,[12,13] later developments take the solution into account.[14] The standard finite-element estimates use Sobolev norms that simultaneously estimate errors in the solution and its derivatives. These estimates might be too conservative because recent finite-volume computations indicate that accurate solutions can be obtained in spite of poor accuracy of gradients.[15–17]

Previously, the authors evaluated the effects of mesh regularity on accuracy of unstructured FVD schemes for various common node-centered and cell-centered schemes.[15,16,18–20] The considered second-order node-centered schemes employ three gradient reconstruction methods: unweighted and weighted least-squares (ULS and WLS, respectively) methods with a linear fit and a reen-auss ( ) method. The following observations concerning relations between accuracy and grid regularity have been made: (1) Convergence and magnitudes of truncation errors are strongly affected by grid regularity and often mislead in predicting convergence and magnitudes of discretization errors. (2) Some common inviscid FVD schemes, e.g., with WLS gradients, produce larger discretization errors (possibly diverging in grid refinement) on almost perfectly regular grids than on very irregular grids with the same degrees of freedom (D F). This striking observation shows the futility of assessing mesh quality independently of the discretization scheme and motivates employment of more stable ULS methods. (3) Convergence and magnitude of discretization errors on isotropic grids are often independent of grid regularity. (4) radient accuracy may degrade on irregular high-aspect-ratio grids effects of this degradation are much stronger on viscous solutions than on inviscid solutions. (5) rid regularity may strongly affect convergence of iterative solvers, e.g., defect-correction iterations. (6) Stochastic tests may be required to account for variations introduced by outlier geometries on irregular grids.

The focus of this paper is on an edge-based node-centered approach. An FVD scheme is considered as edge-based if a loop over edges is sufficient to compute residuals of all equations.[21] Edge-based schemes offer advantages of efficiency (much more efficient than schemes that need to loop over elements in order to compute residuals and linearizations), generality (applicable to agglomeration grids with no explicit elements), and easier grid adaptation. Widely used node-centered FVD schemes[22] are edge-based for inviscid residuals on all grids and for viscous residuals on simplicial grids viscous residuals on non-simplicial elements require an element loop. An attractive feature of an edge-based scheme for integrating fluxes over a median-dual control volume is that the integration is up to third-order accurate on general simplicial grids the integration accuracy may degenerate to first order on general grids including non-simplicial elements.

There is computational evidence that second-order FVD schemes used for practical computations of turbulent flows demonstrate a better accuracy on mixed-element viscous grids with prismatic elements in boundary layers than on fully tetrahedral grids. This evidence is the main motivation for using mixed unstructured grids in spite of efficiency degradation caused by losing the edge-based character of the schemes. Recent publications[23,24] introduced an efficient edge-based FVD scheme using WLS gradient reconstruction with a quadratic fit and showed third-order accuracy for inviscid fluxes on general triangular grids. With this scheme, a comparable or even superior turbulent flow accuracy may be possible on fully tetrahedral grids.

This paper considers effects of mesh regularity on the accuracy of edge-based FVD schemes using ULS gradients computed with a quadratic fit. The inviscid scheme is nominally third-order accurate on general triangular meshes. The viscous scheme is a nominally second-order accurate discretization that uses an average-least-squares method. The schemes have been contrasted with previously studied schemes involving other gradient reconstruction methods such as the reen-auss method and the ULS method with a linear fit.

radient errors and discretization errors are separately studied according to a previously introduced comprehensive methodology.[15,16] A linear convection equation,

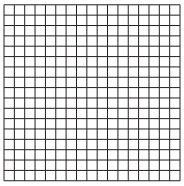
$$(\mathbf{a} \cdot \nabla) U = f, \tag{1}$$

with a velocity vector, $\mathbf{a}$, serves as a model for inviscid fluxes. Poisson's equation

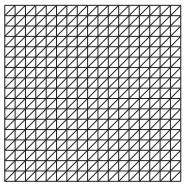
$$\Delta U = f, \tag{2}$$

subject to Dirichlet boundary conditions serves as a model for viscous fluxes. The method of manufactured solutions is used. Solutions are chosen to be smooth on all grids considered, i.e., no accuracy degradation occurs because of a lack of solution smoothness.

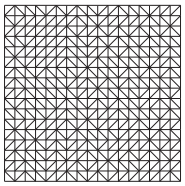
The paper is organized as following. First, grids, FVD schemes, and accuracy measures are briefly described. Then, numerical studies of the FVD accuracy measures are reported for grids of three classes representing isotropic, adapted, and turbulent-flow grids. Finally, conclusions and recommendations are offered concerning the FVD schemes that are expected to be least sensitive to mesh regularity in applications to turbulent flows in complex geometries. Appendix A illustrates high sensitivity of truncation errors to grid regularity. Appendix B presents a study of gradient accuracy as a function of grid deformation typical for curved anisotropic grids used in turbulent-flow computations.



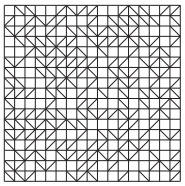
(a) **Type (I): regular quadrilateral grid.**



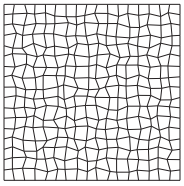
(b) **Type (II): regular triangular grid.**



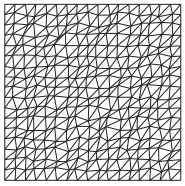
(c) **Type (III): random triangular grid.**



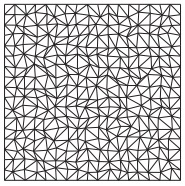
(d) **Type (IV): random mixed grid.**



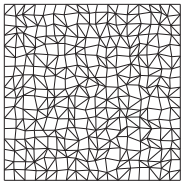
(e) **Type (I_p): perturbed quadrilateral grid.**



(f) **Type (II_p): perturbed triangular grid.**



(g) **Type (III_p): perturbed random triangular grid.**



(h) **Type (IV_p): perturbed random mixed grid.**

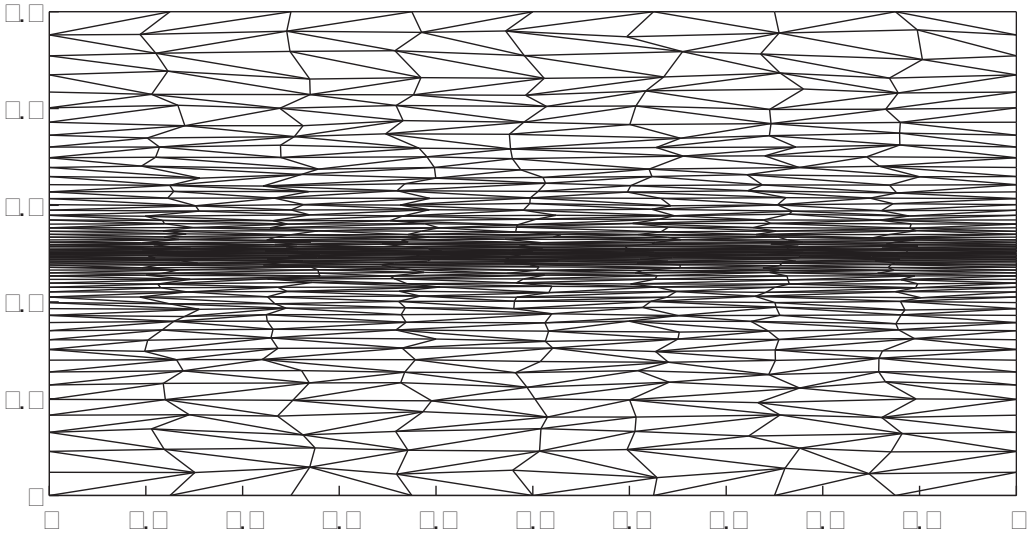
Figure 1. Class A: regular and irregular grids.

II. Grid Classes and Types

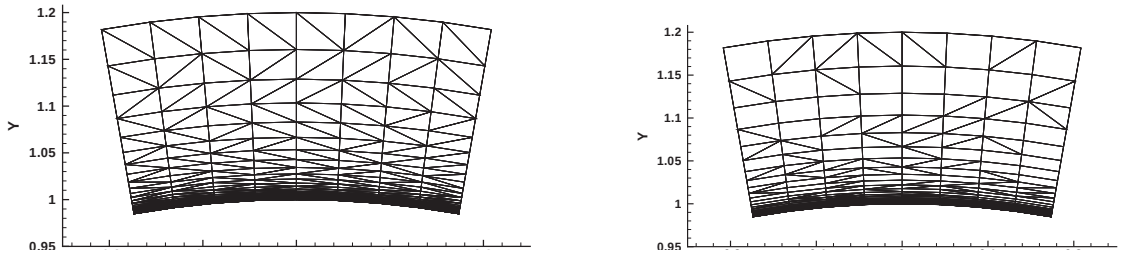
Computational studies are conducted on two-dimensional grids ranging from structured (regular) grids to irregular grids composed of arbitrary mixtures of triangles and quadrilaterals. Highly irregular grids are deliberately constructed through random perturbations of structured grids. Three classes of grids are considered. Class A involves isotropic grids in a rectangular geometry. Class B involves highly anisotropic grids in a rectangular geometry, typical of those encountered in grid adaptation. Class C involves advancing-layer grids varying strongly anisotropically over a curved geometry, typical of those encountered in high-Reynolds number turbulent flow simulations.

Four basic grid types are considered: (I) *regular quadrilateral* (i.e., mapped Cartesian) grids, (II) *regular triangular grids* derived from the regular quadrilateral grids by the same diagonal splitting of each quadrilateral, (III)

random triangular grids, in which regular quadrilaterals are split by randomly chosen diagonals, each diagonal orientation occurring with a probability of half□and (IV) *random mixed-element grids*, in which regular quadrilaterals are randomly split or not split by diagonals□the splitting probability is half□in case of splitting, each diagonal orientation is chosen with probability of half. Nodes of any basic-type grid can be perturbed from their initial positions by random shifts, thus leading to four additional *perturbed* grid types which are designated by the subscript p as (I_p)-(IV_p). The random node perturbation in each dimension is typically defined as $\frac{1}{4}\rho h$, where $\rho \in [-1, 1]$ is a random number and h is the local mesh size along the given dimension. The representative grids of classes A, B, and C are shown in Figures 1, 2, and 3, respectively.



□igure □. □lass □: stretched grid of type (III_p) with 9×65 nodes.



(a) **Grid of type □III□**

(b) **Grid of type □I□□**

□igure □. □lass □: representative 9×33 irregular stretched high-□ grids.

III. □inite-□olume □iscretization □chemes

The FVD schemes are derived from the integral form of a conservation law,

$$\oint_{\partial\Omega} (\mathbf{F} \cdot \hat{\mathbf{n}})\, ds = \int_{\Omega} f d\Omega, \tag{3}$$

where Ω is a control volume with boundary $\partial\Omega$, $\hat{\mathbf{n}}$ is the outward unit normal vector, and ds is the area differential. The general FVD approach requires partitioning the domain into a set of non-overlapping control volumes and numerically implementing Eq. 3 over each control volume.

Node-centered discretization schemes are considered, in which solutions are defined at the primal mesh nodes. The control volumes are constructed around the mesh nodes by the median-dual partition. Node-centered discretization schemes have the same D□F on grids of all types.

For inviscid Eq. 1, the numerical flux,

$$\left(\mathbf{F}^{\mathbf{h}} \cdot \hat{\mathbf{n}}\right) \equiv U^h \left(\mathbf{a} \cdot \hat{\mathbf{n}}\right), \tag{4}$$

at a control-volume boundary is computed according to the flux-difference-splitting scheme,[26]

$$U^h \left(\mathbf{a} \cdot \hat{\mathbf{n}}\right) = \frac{1}{2} \left(U_L + U_R\right) \left(\mathbf{a} \cdot \hat{\mathbf{n}}\right) - \frac{1}{2} \left|\left(\mathbf{a} \cdot \hat{\mathbf{n}}\right)\right| \left(U_R - U_L\right), \tag{5}$$

where the first and second terms represent the flux average and the dissipation, respectively□U_L and U_R are the "left" and "right" solutions reconstructed at the edge midpoint by using solutions and gradients defined at the nodes connected by the edge. The edge-based flux integration scheme approximates the integrated flux through the two faces linked at the edge midpoint by $U^h \left(\mathbf{a} \cdot \mathbf{n}\right)$, where $\mathbf{n}$ is the combined directed-area vector of the adjacent faces.

The integration scheme is computationally efficient. For exact fluxes, the integration scheme provides third-order accuracy on regular simplicial grids of type (II), second-order accuracy on regular quadrilateral and general simplicial grids of types $(I), (III), (II_p)$, and (III_p), and first-order accuracy on mixed-element and perturbed quadrilateral grids of types $(IV), (IV_p)$, and (I_p).[18,19,27]

It was shown[23,24] that third order discretization accuracy is achieved on simplicial grids with WLS□ gradients employing a quadratic fit. Third-order accuracy on simplicial grids has been confirmed with quadratic-fit ULS□ gradients used herein. Note that five neighbors are typically sufficient for a quadratic fit. □n triangular grids considered in this study, the average number of edge-connected neighbors is six□and the minimum number of edge-connected neighbors for an interior node on any grid is four. In cases when the least-squares stencil of the nearest edge-connected neighbors is not sufficient for a quadratic fit, the stencil is expanded to include neighbors of neighbors.

For viscous Eq. 2, the numerical flux is defined as

$$\left(\mathbf{F}^{\mathbf{h}} \cdot \hat{\mathbf{n}}\right) \equiv \left(\nabla^{\mathbf{r}} U \cdot \hat{\mathbf{n}}\right), \tag{6}$$

where $\nabla^{\mathbf{r}} U$ is the gradient reconstructed at the face of the control volume. Two gradient reconstruction schemes are considered. First, the averaged least-squares (Avg-LS□) scheme averages the ULS□ gradients at the nodes to compute the face gradient.[28,29] Second, the □□ scheme[15,22] computes gradients at the primal elements and uses them in face-gradient computations at control-volume boundaries. The □□ scheme is widely used in node-centered codes and equivalent to a □alerkin finite-element (linear-element) discretization for triangular□tetrahedral grids. Both schemes use the edge gradient to augment the face gradient and increase the h-ellipticity[30] of the diffusion operator[15,21] and thus, avoid checkerboard instabilities. The gradient augmentation is introduced in the face-tangent form.[29] Note that when the edge is normal to the face, the edge gradient is the only contributor to the flux. For the □□ scheme, the implementation of gradient augmentation on three-dimensional non-simplicial grids requires looping over elements and thus, alters the edge-based character of the scheme. The augmentation does not affect the face gradient within a simplex element and thus, the □□ scheme is edge based on simplicial grids. Both Avg-LS□ and □□ schemes possess second-order accuracy for viscous fluxes on general mixed-element grids.[18,19,28,29]

I□. □ccuracy □ easures

The accuracy is analyzed for known exact or manufactured solutions. The forcing function and boundary values are found by substituting this solution into the governing equations, including boundary conditions. The discrete forcing function is defined at the nodes that are not necessarily located at centroids of control volumes. Boundary conditions are over-specified, i.e., discrete solutions at boundary control volumes and, possibly, at their neighbors are specified from the manufactured solution. Unless described otherwise, the figures in this paper show accuracy measures versus an effective meshsize which is computed as the L_1 norm of the $\sqrt{V}$ function, where V is a measure of the control volume,

$$V = \int_{\Omega} d\Omega. \tag{7}$$

Relations between different methods of computing the effective meshsize are discussed in Ref. 19.

IV.A. Discretization error

The main accuracy measure is the *discretization error*, E_d, which is defined as the difference between the exact discrete solution, U^h, of the discretized Eq. 3 and the exact continuous solution, U, to the corresponding differential equations,

$$E_d = U - U^h, \tag{8}$$

where U is sampled at mesh nodes.

IV.B. Accuracy of gradient reconstruction

The accuracy of the gradient approximation is also important. The gradient reconstruction accuracy is evaluated by comparing the reconstructed gradient, $\nabla^{\mathbf{r}} U$, with the exact gradient, ∇U. The accuracy of a ULSQ gradient is evaluated by comparing the reconstructed and exact gradients at nodes. The accuracy of a GG gradient is evaluated at element centers computed as the average of the corresponding element vertexes. The error in the gradient reconstruction is measured as

$$E_{\mathrm{g}} = |\nabla^{\mathbf{r}} U - \nabla U|. \tag{9}$$

V. Class A: Isotropic Grids in Rectangular Geometry

V.A. Grid and solution specifications

Sequences of consistently refined[19] grids with $5^2, 9^2, 17^2, 33^2, 65^2, 129^2$, and 257^2 nodes are generated on the unit square $[0,1] \times [0,1]$. Irregularities are introduced at each grid independently, so the grid metrics remain discontinuous on all irregular grids. With the random perturbation range limited by a quarter of the local mesh size, the angles of triangular elements can approach 180° and the ratio of the neighboring cell volumes can be arbitrarily high.

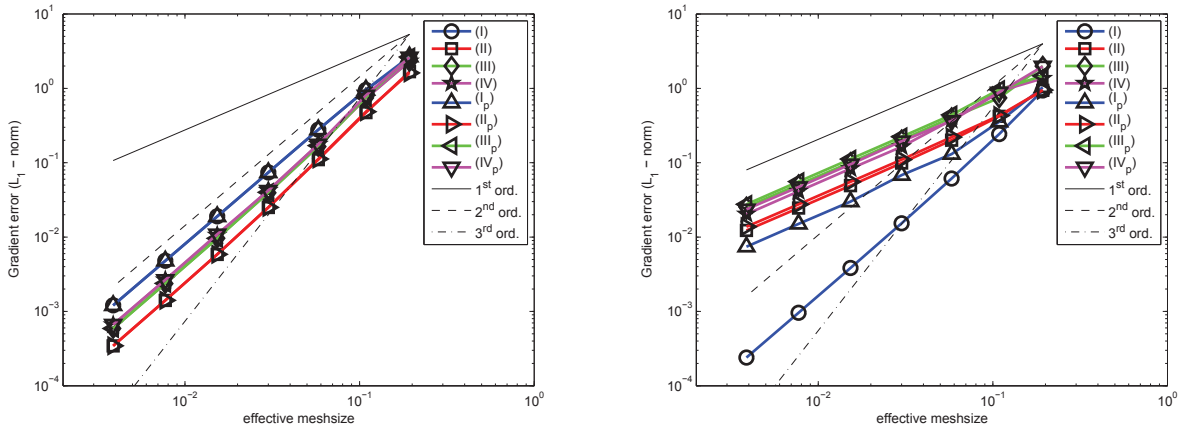
The exact solution is $U = \sin(\pi x - 2\pi y)$, so for the inviscid Eq. 1 with $\mathbf{a} = (2,1)$, the force, f, is zero, and for the viscous Eq. 2, $f = -5\pi^2 \sin(\pi x - 2\pi y)$. The boundary conditions are over-specified from the manufactured solution for all nodes linked to the boundary.

V.B. Gradient reconstruction errors

Figure 4 shows the variation of the L_1 norm of the gradient error. As expected, the ULSQ gradient reconstruction with a quadratic fit is second-order accurate on all grids. The GG gradient reconstruction is second order only on perfect grids of type (I); on all other grids, the GG gradients are first-order accurate. All equivalent-order methods provide

very similar errors. Thus, no mesh regularity effects are observed for the L_1 norm of the gradient error on isotropic grids.

Although not shown, the observed L_∞ norms of the gradient errors converge with the same orders as the corresponding L_1 norms, but the L_∞ norms of GG gradient error on grids of types (III_p) and (IV_p) are an order of magnitude greater than the L_∞ norms of other first-order errors. The latter effect is caused by gradient accuracy deterioration on triangular elements with obtuse angles approaching $180°$. Theoretically, with an infinitesimal probability, the GG gradient error may become infinitely large at an element with a vanishing volume. As opposed to the anisotropic grids considered below, elements with extremely obtuse angles occur infrequently and in isolation on isotropic grids. Thus, discretization errors are not affected.



(a) Quadratic fit at node (b) GG at element

Figure 5. Accuracy of gradient reconstruction on isotropic grids. Manufactured solution is $U = \sin(\pi x - 2\pi y)$.

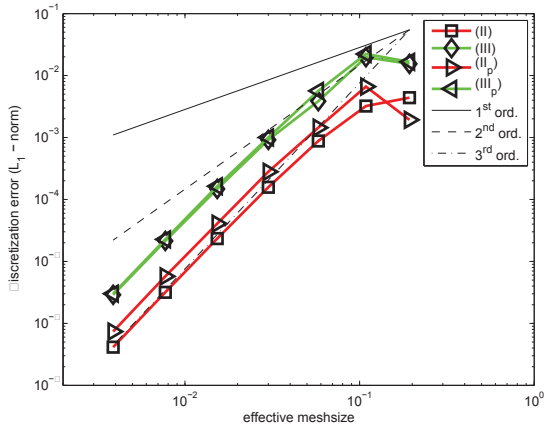
V.C. Discretization errors

Convergence rates of the L_1 norm of discretization errors for inviscid and viscous fluxes are shown in Figures 5 and 6, respectively. This is an example where inviscid accuracy on simplicial meshes is superior to that on meshes with quadrilateral elements. This is not a surprise because the inviscid scheme used in this study is designed to be third order only on simplicial grids.[23,24] The edge-based integration scheme used in this scheme is known to deteriorate to first order on grids of types (I_p), (IV), and (IV_p).[18,19,27] On triangular grids, the discretization accuracy of inviscid solutions is not sensitive to mesh regularity. If anything, discretization errors are somewhat smaller on topologically structured grids of types (II) and (II_p). Discretization errors for viscous fluxes show no sensitivity to mesh regularity. The errors for both Avg-LSQ and GG schemes are practically identical to the plotting accuracy for all grids.

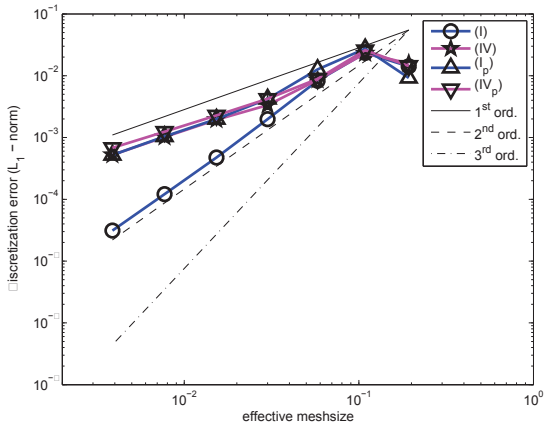
VI. Class B: Anisotropic Grids in Rectangular Geometry

VI.A. Grid and solution specifications

This section considers FCC schemes on stretched grids generated on rectangular domains. Figure 2 shows an example grid with the maximal aspect ratio $\mathcal{A} = 1,000$. A sequence of consistently refined stretched grids is generated on the rectangle $(x, y) \in [0, 1] \times [0, 0.5]$ in the following 3 steps.

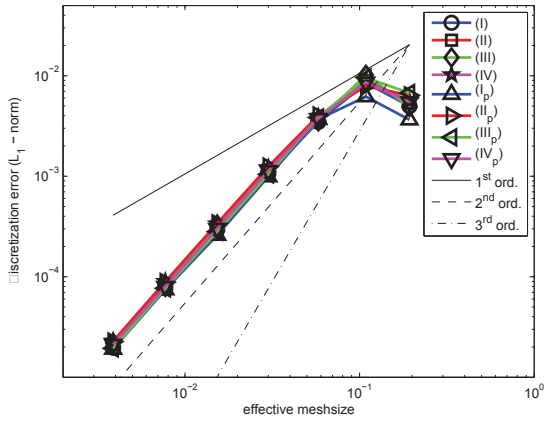


(a) □riangular mes□es

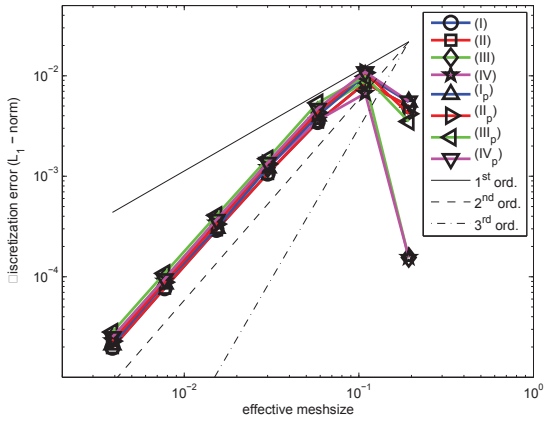


(b) □ i□ed and □uadrilateral mes□es

□igure □. In□iscid discretization errors on isotropic grids. □ anufactured solution is $U = \sin(\pi x - 2\pi y)$.



(a) A□g□□□



(b) GG

□igure □. Viscous discretization errors on isotropic grids. □ anufactured solution is $U = \sin(\pi x - 2\pi y)$.

1. A background regular rectangular grid with $N = (N_x + 1) \times (N_y + 1)$ nodes and the horizontal mesh spacing $h_x = 1/N_x$ is stretched toward the horizontal line $y = 0.25$. The y-coordinates of the horizontal grid lines in the top half of the domain are defined as

$$y_{\frac{N_y}{2}+1} = 0.25; \qquad y_j = y_{j-1} + \hat{h_y}\beta^{j-\left(\frac{N_y}{2}+1\right)}, \qquad j = \frac{N_y}{2} + 2, \ldots, N_y, N_y + 1. \tag{10}$$

□ere $\hat{h_y} = h_x/A$ is the minimal mesh spacing between the vertical lines, $\mathcal{A} = 1,000$ is a fixed maximal aspect ratio, and β is a stretching factor which is found from the condition $y_{N_y+1} = 0.5$. The stretching in the bottom half of the domain is defined analogously.

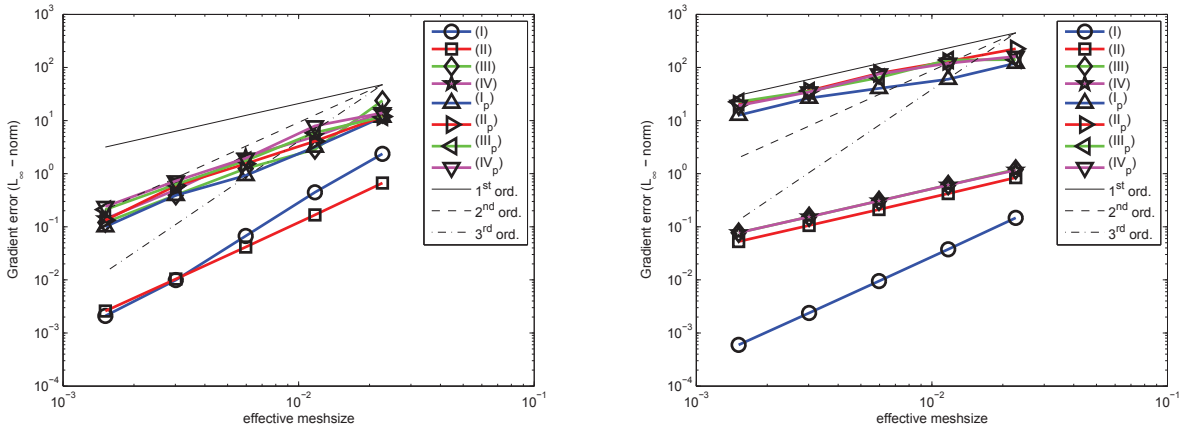
2. Irregularities are introduced by random shifts of interior nodes in the vertical and horizontal directions. The vertical shift is defined as $\Delta y_j = \frac{3}{16}\rho \min(h_y^{j-1}, h_y^j)$, where ρ is a random number between -1 and 1, and h_y^{j-1} and h_y^j are vertical mesh spacings on the background stretched mesh around the grid node. The horizontal shift is introduced analogously, $\Delta x_i = \frac{3}{16}\rho h_x$. With these random node perturbations, all perturbed quadrilateral cells are convex.

3. Each perturbed quadrilateral is randomly triangulated with one of the two diagonal choices; each choice occurs with a probability of one half.

Sequences of consistently refined stretched grids with maximum aspect ratio $\mathcal{A} = 1,000$ including $9 \times 65, 17 \times 129, 33 \times 257$, 65×513, and 129×1025 nodes have been considered. The corresponding stretching ratios are $\beta \approx 1.207, 1.098, 1.048, 1.025$, and 1.012. The aspect ratio near the external horizontal boundaries is about 2.7.

In the tests on grids of lass performed with either the manufactured solution $\sin(\pi x - 2\pi y)$ or extended over-specification used in tests on grid of lass A, the asymptotic behavior of the discretizations errors for viscous uxes was not observed on coarse grids. The exhibited discretization errors were uncharacteristically low on coarse grids, but did not converge with the asymptotic order. The discretization errors for this specific manufactured solution on the chosen domain are small in the interior and peak toward the boundary. Thus, over-specification that involves all neighbors of boundary nodes affects solutions on a too large portion of stretched grids. As a result, the manufactured solution has been changed to $U = \cos(\pi x - 2\pi y)$; the discretization errors for this solution peak in the middle of the computational domain. Also only solutions at boundary nodes are over-specified, and not at their neighbors as was done for lass A grids. With these changes, the asymptotic behavior of the discretizations errors for the viscous uxes is established on relatively coarse grids. ote that the forcing term for inviscid equations is still $f = 0$ for $\mathbf{a} = (2, 1)$.

VI.B. Gradient reconstruction errors



(a) uadratic t at node (b) GG at element

igure Accuracy of gradient reconstruction on stretched grids it ma imum aspect ratio $A = 1,000$. anufactured solution is $U = \cos(\pi x - 2\pi y)$.

A recent study[20] assessed the accuracy of gradient approximations on various grids with high aspect ratio $\mathcal{A} = \frac{h_x}{h_y} \gg 1$. The study indicates that for rectangular geometries and functions predominantly varying in the direction of

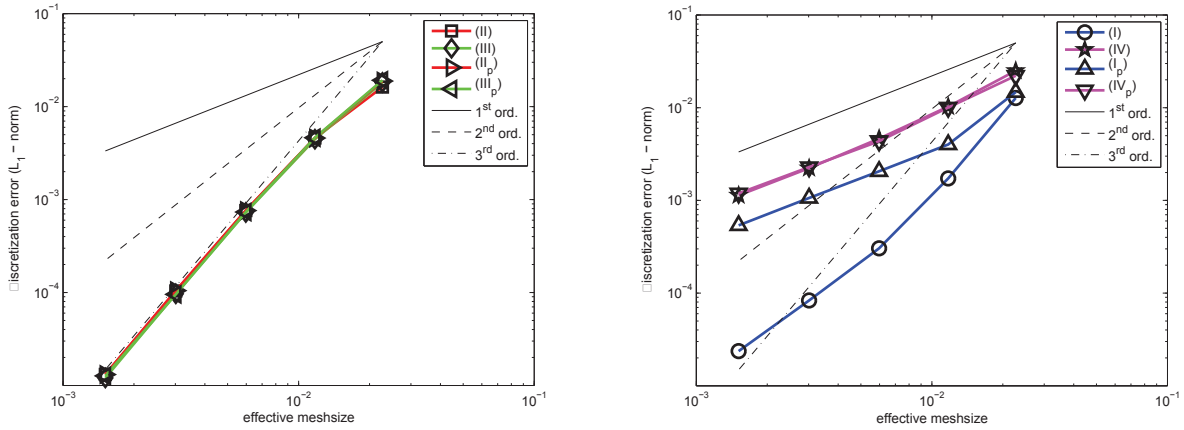
small mesh spacing (y-direction here), gradient reconstruction is accurate and provides small relative error while converging with at least first order in consistent refinement on grids of all types. For manufactured solutions significantly varying in the direction of larger mesh spacing (x-direction), the gradient reconstruction may produce extremely large relative errors $O(\mathcal{A}h_x^p)$ affecting the accuracy of the y-directional gradient component. □ere, p is the formal gradient reconstruction order; $p = 1$ for the GG method and for the ULSQ method with a linear fit; $p = 2$ for the ULSQ scheme with a quadratic fit.

A summary of the results concerned with gradient accuracy on anisotropic grids is presented in Table 1. The gradient is accurately reconstructed on all unperturbed grids by the GG scheme. All gradient reconstruction methods considered may generate large relative errors on perturbed grids of types $(I_p) - (IV_p)$.

□a□le □. Relati□e error of gradient reconstruction on anisotropic grids for solutions □it□ signi□cant □ariation in t□e x-□direction of larger mes□ spacing.

Grid Types	(I)	(II)	(III)	(IV)	$(I_p) - (IV_p)$
ULSQ-linear fit at node	$O(h_x^2)$	$O(h_x^2)$	$O(\mathcal{A}h_x)$	$O(\mathcal{A}h_x)$	$O(\mathcal{A}h_x)$
ULSQ-quadratic fit at node	$O(h_x^2)$	$O(h_x^2)$	$O(\mathcal{A}h_x^2)$	$O(\mathcal{A}h_x^2)$	$O(\mathcal{A}h_x^2)$
GG at element center	$O(h_x^2)$	$O(h_x)$	$O(h_x)$	$O(h_x)$	$O(\mathcal{A}h_x)$

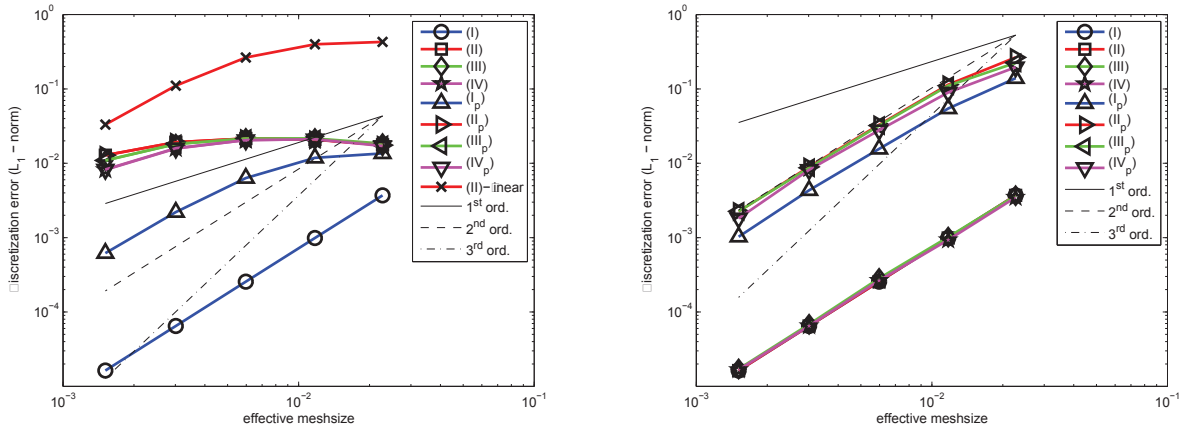
The convergence of the L_∞ norm of gradient errors is shown in Figure 7. The L_∞ norm is used to highlight the worst gradients observed in high-aspect ratio regions of the stretched grids of □lass □. All quadratic-fit ULSQ gradients converge with second order, but the magnitude of the gradient errors is sensitive to grid regularity. As shown in Table 1, with any deviation from the regularity of grids of types (I) and (II), the ULSQ gradient error becomes proportional to aspect ratio. The GG gradients converge with first order on all grids beside the grids of type (I), where a second-order convergence is observed. In spite of a lower order convergence, the GG gradients show a clear advantage over the ULSQ gradients on coarse unperturbed grids of types (I)-(IV). The GG scheme on such grids provides gradient accuracy independent of aspect ratio. □n perturbed grids of types (I_p)-(IV_p), the GG errors are also proportional to the aspect ratio, and quadratic-fit ULSQ gradients are preferable.



(a) □riangular mes□es

(b) □ i□ed and □uadrilateral mes□es

□igure □. In□iscid discretization errors on anisotropic stretc□ed grids □it□ ma□imum aspect ratio $\mathcal{A} = 1,000$. □anufactured solution is $U = \cos(\pi x - 2\pi y)$.



(a) A□g□□□ (b) **GG**

□igure □. Viscous discretization errors on anisotropic stretc□ed grids □it□ ma□imum aspect ratio $\mathcal{A} = 1,000$. □anufactured solution is $U = \cos(\pi x - 2\pi y)$.

VI.C. Discretization errors

The convergence of the L_1 norm of discretizations errors for inviscid □uxes is shown in Figure 8. The convergence characteristics are similar to those exhibited on isotropic grids of □lass A. Third-order convergence insensitive to grid regularity is observed on all triangular grids. □onvergence on grids of type (I) is second order, but any irregularity on mixed and quadrilateral meshes degrades the convergence to first order.

The convergence of the L_1 norm of discretization errors for viscous □uxes is shown in Figure 9. All discretization errors converge with second order. While second-order convergence of the Avg-LSQ scheme is not apparent in Figure 9(a) on triangular and mixed-element grids, a second-order slope has been attained on finer grids. For reference, convergence of the errors obtained with a linear fit on grids of type (II) is also shown. The Avg-LSQ errors are relatively small only on pure quadrilateral grids of types (I) and (I_p). The magnitude of errors obtained with a quadratic fit is much smaller than the magnitude of errors obtained with a linear fit. □owever, discretization errors of the GG scheme are significantly better than any of the Avg-LSQ errors. The GG errors are clearly divided into two groups. The errors on unperturbed grids of types $(I) - (IV)$ are small on all grids; the errors on perturbed grids are roughly two orders of magnitude higher for any given number of □□F. The ratio is about the same as the ratio between gradient errors shown in Figure 7(b).

VII. Class C: Grids □it□ Cur□ature and □ig□ Aspect Ratio

VII.A. Grid and solution specifications

In this section, we discuss F□□ schemes on grids with curvature and high aspect ratio. The grid nodes are generated from a cylindrical mapping, where (r, θ) denote polar coordinates with spacings of h_r and h_θ, respectively. The grid aspect ratio is defined as the ratio of mesh sizes in the circumferential and the radial directions, $\mathcal{A} = Rh_\theta/h_r$, where R is the radius of curvature.

The curvature-induced mesh deformation parameter Γ[1□,16] is defined as□

$$\Gamma = \frac{R\,(1-\cos(h_\theta))}{h_r} \approx \frac{Rh_\theta^2}{2h_r} = \mathcal{A}\frac{h_\theta}{2}. \tag{11}$$

The following assumptions are made about the range of parameters□$R \approx 1$, $\mathcal{A} \gg 1$, and $\Gamma h_r \ll 1$, which implies that both h_r and h_θ are small. For a given value of $\mathcal{A}$, the parameter Γ may vary□$\Gamma \ll 1$ indicates meshes that are locally (almost) non-deformed. As a practical matter, grids with $\Gamma < 0.2$ can be considered as nominally non-curved. In a mesh refinement that keeps $\mathcal{A}$ fixed, $\Gamma = O(\mathcal{A}h_\theta)$ asymptotes to zero. This property implies that on fine enough grids with fixed curvature and aspect ratio, the error convergence is expected to be the same as on similar □lass □ grids generated on rectangular domains with no curvature.

Four basic types of grids are studied in the cylindrical geometry. Unlike □lass □ grids used in the rectangular geometry, random node perturbation is not applied to high-Γ grids of □lass □ because even small perturbations in the circumferential direction may lead to non-physical control volumes. Representative stretched grids of types (III) and (IV) are shown in Figure 3.

The manufactured solution considered in this section is $U = \sin(5\pi r)$. The convection direction is changed to a variable tangential direction $\mathbf{a} = (y/r^2, -x/r^2)$, so the inviscid forcing term remains zero. Solutions at boundary nodes are over-specified.

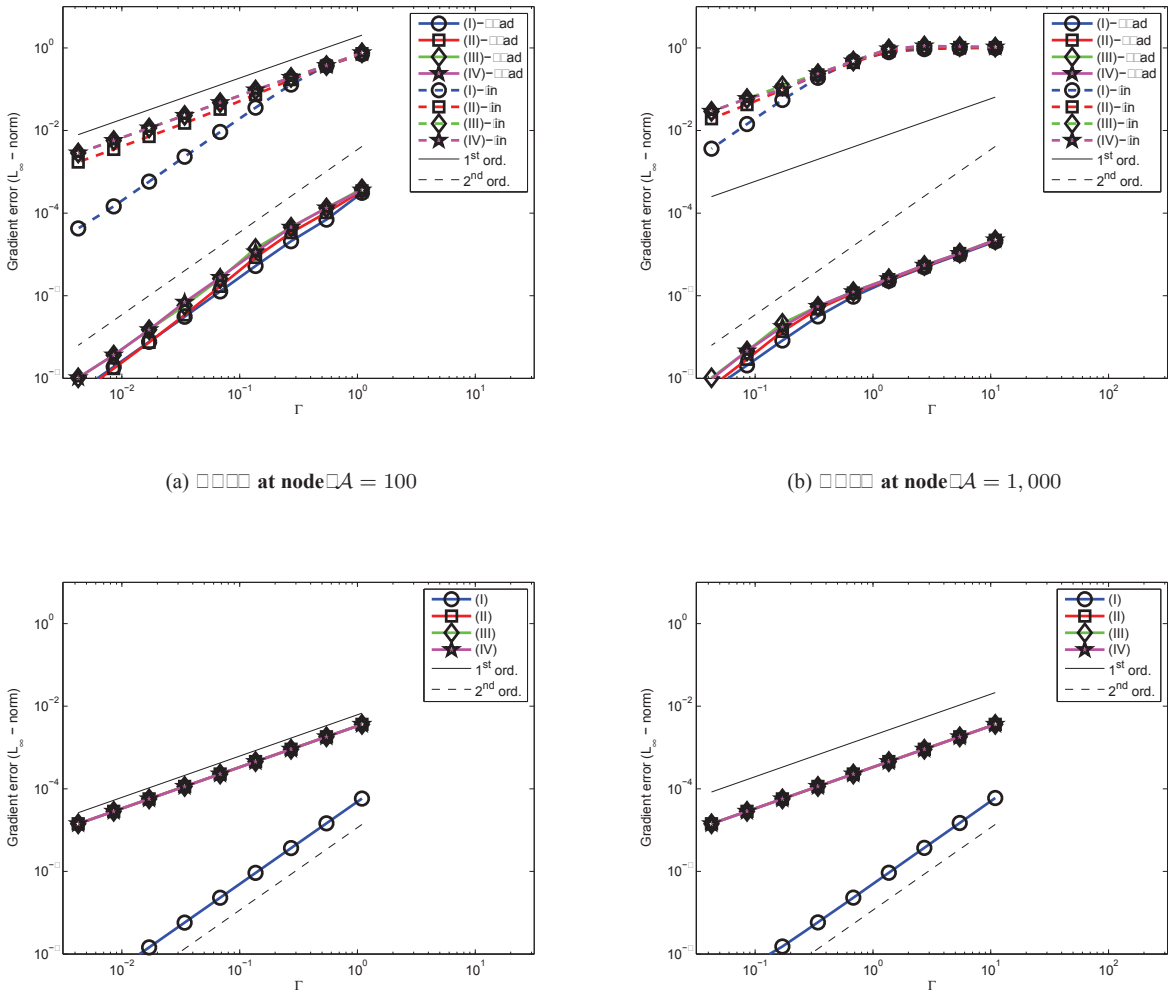
VII.B. Gradient reconstruction errors

□a□le □. Relati□e errors of gradient reconstruction for manufactured solutions □arying only in t□e radial direction on □ig□□Γ grids.

Grid Types	(I)	(II)	(III)	(IV)
ULSQ-linear fit	$O(1)$	$O(1)$	$O(1)$	$O(1)$
ULSQ-quadratic fit	$O(h_\theta)$	$O(h_\theta)$	$O(h_\theta)$	$O(h_\theta)$
GG	$O(h_\theta^2)$	$O(h_\theta)$	$O(h_\theta)$	$O(h_\theta)$

□ur main interest is solutions varying predominantly in the radial direction on grids with $\Gamma \gg 1$ corresponding to meshes with large curvature-induced deformation. The errors of gradient reconstruction for a radial solution are summarized in Table 2. The ULSQ gradient approximation with a linear fit is zeroth-order accurate for such solutions, in agreement with computations and analysis reported earlier.[17,2□] The use of the ULSQ method with a quadratic fit dramatically improves gradient accuracy on high-Γ grids leading to a first-order convergence of gradient errors on grids with high Γ.

The computational tests are performed with downscaling[19,20] on a sequence of narrow arc-shaped domains with the angular extent of $\frac{\pi}{9}L$ radians and the radial extent of $1 \le r \le 1 + \frac{\pi}{9}L\mathcal{A}^{-1}$. The scale L changes as $L = 2^{-n}, n = 0, \ldots, 8$. □n each domain, a 17×17 grid is generated with nodes uniformly spaced in the polar coordinates. Figure 10 shows convergence of the L_∞ norms of gradient errors computed for the manufactured solution $U = \sin(5\pi r)$ on grids with aspect ratios $\mathcal{A} = 100$ and $\mathcal{A} = 1,000$. The errors are shown versus the grid deformation parameter, Γ, defined in Eq. 11. Figures 10(a) and 10(b) show convergence of ULSQ gradient errors computed with quadratic and linear fits on grids of types (I)□(IV). Figures 10(c) and 10(d) show convergence of GG gradient errors. As known from previous studies,[1□□17] the errors of GG gradients are small and show the order property on all grids. The ULSQ gradients computed with a linear fit lose accuracy on high-Γ grids. The ULSQ gradients computed with a quadratic fit recover a first-order convergence on high-Γ grids and show the smallest error magnitudes on grids of types (II), (III), and (IV). The GG gradients show the smallest errors on regular quadrilateral grids of type (I). Appendix □ presents a detailed study of gradient reconstruction errors for ULSQ methods with linear and quadratic fits on a family of stencils corresponding to a wide range of Γ.



(a) □□□□ **at node**□$\mathcal{A} = 100$

(b) □□□□ **at node**□$\mathcal{A} = 1,000$

(c) **GG at element**□$\mathcal{A} = 100$

(d) **GG at element**□$\mathcal{A} = 1,000$

□igure □□. Accuracy of gradient reconstruction on □ig□□ grids. □ anufactured solution is $U = \sin(5\pi r)$.

VII.C. Discretization errors

□omputational grids used in the grid-refinement study of discretization errors are radially stretched grids with a radial extent of $1 \leq r \leq 1.2$ and an angular extent of $20°$. Fixed maximal aspect ratios are used. The maximal aspect ratio is $\mathcal{A} \approx 1,000$ for viscous computations. The grids have four times more cells in the radial direction than in the circumferential direction. The maximum value of Γ changes approximately as $\Gamma \approx 22, 11, 5.5, \ldots$. The corresponding grid stretching ratios change as $\beta = 1.25, 1.11, 1.06, \ldots$.

The third-order inviscid scheme produces highly accurate solutions, so local errors become very small on relatively coarse highly stretched grids and convergence is obscured by round-off errors interfering with the solutions. A reduced maximal aspect ratio of $\mathcal{A} \approx 100$ has been chosen for inviscid computations.

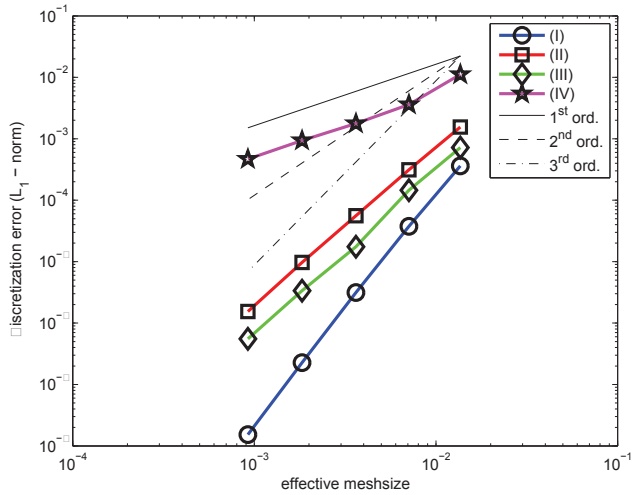
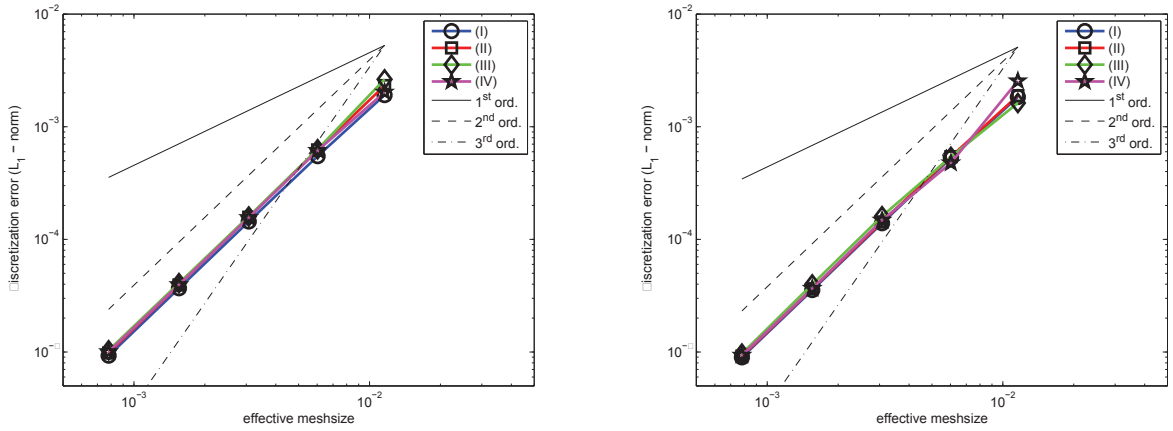


Figure 11. Inviscid discretization errors on Figure 7 stretched grids with maximum aspect ratio $\mathcal{A} = 100$. Manufactured solution is $U = \sin(5\pi r)$.



(a) Agard0012 (b) GG

Figure 12. Viscous discretization errors on Figure 7 stretched grids with maximum aspect ratio $\mathcal{A} = 1,000$. Manufactured solution is $U = \sin(5\pi r)$.

Convergence of the L_1 norm of discretization errors is shown in Figures 11 and 12 for inviscid and viscous fluxes, respectively. The inviscid errors converge with (almost) fourth order on grids of type (I), with third order on grids of types (II) and (III), and with first order on grids of type (IV). The unusually high order of convergence on grids of type (I) is explained by the fact that, for a manufactured solution varying in the radial direction only, the inviscid scheme on grids of type (I) turns into a fourth-order pure one-dimensional scheme. Any solution variation in the circumferential direction results in the expected second-order convergence on grids of type (I). Note that, because of asymmetric gradient-reconstruction stencil on grids of types (II) and (III), the scheme does not become one-dimensional and thus, its third order of convergence on these grids is independent of solution variation. Second-order

convergence and no sensitivity to grid type are observed for both viscous schemes.

VIII. Conclusions

The effects of mesh regularity on the accuracy of unstructured node-centered finite-volume discretizations for viscous and inviscid uxes have been considered for an edge-based approach that use unweighted least-squares gradient reconstruction with a quadratic fit. The inviscid scheme is nominally third-order accurate on general triangular meshes.[23,24] The viscous scheme is a nominally second-order accurate discretization that uses an average-least-squares method with a face-tangent augmentation.[28,29] The results have been contrasted with previously studied schemes involving other gradient reconstruction methods such as the Green-Gauss method and the unweighted least-squares method with a linear fit. Gradient errors, truncation errors, and discretization errors have been separately studied according to a previously introduced methodology.[1,16]

The methodology considers three classes of grids lass A includes isotropic grids in a rectangular geometry, lass includes anisotropic grids representative of adaptive-grid simulations, and lass includes anisotropic advancing-layer grids representative of high-Reynolds number turbulent ow simulations over a curved body. Regular and irregular grids have been considered, including mixed-element grids and grids with random perturbations of nodes. Grid perturbations and stretching have been introduced independently of solution variation to bring out the worst possible behavior.

The gradient accuracy deteriorates on high-aspect-ratio perturbed grids. n grids of lass , the gradient errors converge with the design orders first order for the Green-Gauss method and the least-squares method with a linear fit and second order for the least-squares method with a quadratic fit. The least-squares gradient errors become proportional to the aspect ratio on all irregular grids. n grids with node perturbation, all gradient errors are proportional to the aspect ratio. n lass grids characterized by a high deformation parameter Γ, the Green-Gauss gradient errors converge with at least first order and are small on all grids. The errors of least-squares gradients with a quadratic fit converge with first order. The magnitude of the quadratic-fit errors is superior to the $O(1)$ magnitude observed with a linear fit.

As observed previously[8,11,19] and confirmed here in Appendix A, lack of mesh regularity strongly affects truncation errors, which converge with lower-than-design order on all irregular meshes. iscous truncation errors do not converge at all on perturbed grids.

Inviscid discretization errors are practically insensitive to mesh regularity on triangular grids, demonstrating a third-order convergence and small variation of the error magnitudes. iscretization accuracy is more sensitive to mesh regularity on grids with quadrilateral elements. n those grids, the results observed with the least-squares method with a quadratic fit show no advantage over previous results obtained with a linear fit,[16,19] both showing first-order convergence on mixed and perturbed quadrilateral grids.

In all cases, the viscous discretization errors asymptotically converge with second order. Similar to the gradient accuracy, the magnitude of discretization errors of viscous solutions is insensitive to grid regularity on grids of lass A, but may be sensitive on grids of classes and . n such grids, the Green-Gauss method is the most accurate, although the errors on the grids with node perturbation are still significantly larger than errors on grids with unperturbed nodes. Asymptotically, the difference is proportional to the aspect ratio. Accuracy of the average-least-squares methods deteriorates on irregular high-aspect-ratio grids, although the deterioration is less with a quadratic fit than with a linear fit.

The following recommendations are offered

1. The unweighted least-squares method with a quadratic fit is highly recommended as a robust way to compute accurate gradients on all grids.

2. The edge-based scheme that uses the unweighted least-squares method with a quadratic fit is recommended for inviscid uxes. n triangular grids, it produces third-order accurate solutions and is insensitive to mesh regularity.

3. The Green-Gauss scheme is recommended for viscous □uxes. □n isotropic and advanced-layer grids of classes A and □, both Green-Gauss and averaged-least-squares methods produce uniformly second-order solutions and are insensitive to mesh regularity. □n grids of □lass □, there is a sensitivity to grid regularity; the Green-Gauss solutions are less sensitive than averaged-least-squares solutions.

Robust iterative convergence is also critically important for practical applications. The solver for the third-order scheme reported previously[23] failed to converge on high-Γ grids of □lass □. This failure is attributed in part to use of a WLSQ gradient reconstruction that causes difficulties for iterative solvers in complex geometries.[2□] Although, we do not consider iterative convergence in this paper, preliminary tests indicate that a combination of a ULSQ method with an approximate mapping technique[1□,16] enables fast and robust convergence of defect-correction iterations for this third-order scheme on high-aspect-ratio grids in complex geometries. Also, the approximate-mapping approach to gradient reconstruction can recover a second-order convergence of gradient errors on high-Γ grids of □lass □.

The overall conclusion is that relations between mesh characteristics and solution accuracy are complicated. The mesh regularity affects gradient, truncation, and discretization errors in dramatically different ways. The resolution is expected in the form of ad□oint-based grid adaptation that directly and rigorously connects the local mesh properties with the desired solution outcome.

A. □runcation errors

Truncation error, E_t, characterizes the accuracy of approximating the differential equations. For finite differences, the truncation error is defined as the residual obtained after substituting the exact solution U into the discretized differential equations.[31] For F□□ schemes, the traditional truncation error is usually defined from the time-dependent standpoint.[32,33] In the steady-state limit, it is defined (e.g., in Ref. 34) as the residual computed after substituting U into the normalized discrete Eq. 3,

$$E_t = \frac{1}{V}\left[-\int_{\Omega} f^h \, d\Omega + \oint_{\partial\Omega} \left(\mathbf{F^h}\cdot\hat{\mathbf{n}}\right)\, ds \right], \tag{12}$$

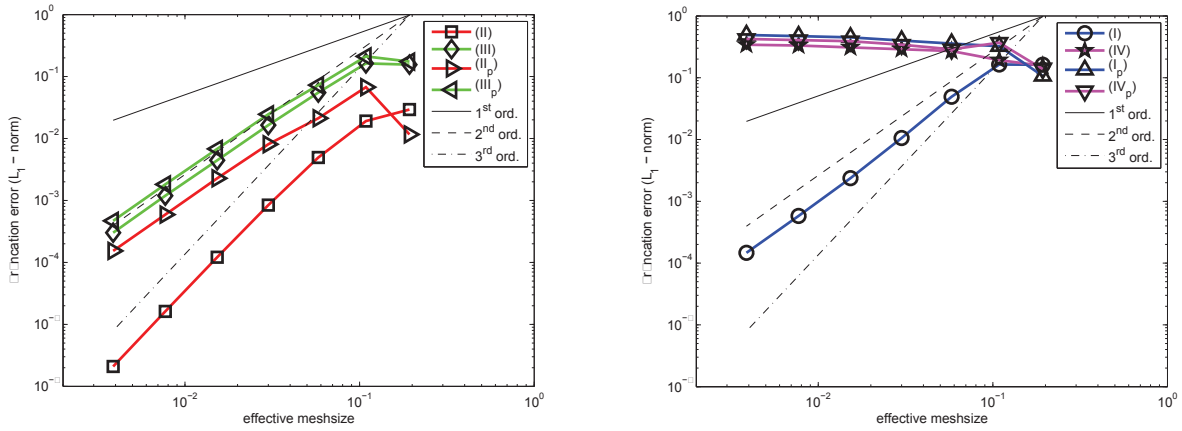
where V is the measure of the control volume, Eq. 7, f^h is an approximation of the forcing function f on Ω, and the integrals are computed according to quadrature formulas.

The truncation errors are extremely sensitive to mesh regularity. □onvergence rates of the L_1 norm of truncation errors for inviscid and viscous □uxes on isotropic grids of □lass A are shown in Figures 13 and 14, respectively. The inviscid scheme and the viscous Avg-LSQ scheme use the ULSQ method with a quadratic fit; the viscous GG scheme is shown for comparison. The grids and manufactured solution are defined in Section □.A.

The inviscid errors converge with third order only on regular triangular meshes of type (II). □n irregular triangular grids of types (III), (II_p), and (III_p) and on perfect quadrilateral grid of type (I), the inviscid truncation errors converge with second order. Irregularities on grids with quadrilateral elements (types (IV), (I_p), and (IV_p)) lead to zeroth-order convergence.

Similar sensitivity is observed for the truncation errors of viscous □uxes discretized by the Avg-LSQ scheme with second-order accurate ULSQ gradients (Figures 14(a) and 14(b)). The second-order convergence is observed only on perfectly regular grids of types (I) and (II). The convergence deteriorates to first order on irregular triangular grids and to zeroth order on mixed-element and perturbed quadrilateral grids. For viscous □uxes discretized with the GG scheme (Figures 14(c) and 14(d)), truncation errors do not converge on any but perfectly regular grids of types (I) and (II). □ote that GG scheme produces identical discretizations on grids of types (I), (II), and (III).[1□] Thus, corresponding GG solutions and truncation errors on grids of types (I) and (II) are always identical. □ifferent results on grids of type (III) are explained by the differences in the dual volumes.

The qualitative behavior (orders of convergence) of truncation errors on anisotropic grids of □lass □ is the same as on isotropic grids, shown in Figures 13 and 14. □n grids with similar □□F, the magnitude of the errors increases



(a) Triangular grids

(b) Mixed and Quadrilateral grids

Figure 11. Inviscid truncation errors on isotropic grids. Manufactured solution is $U = \sin(\pi x - 2\pi y)$.

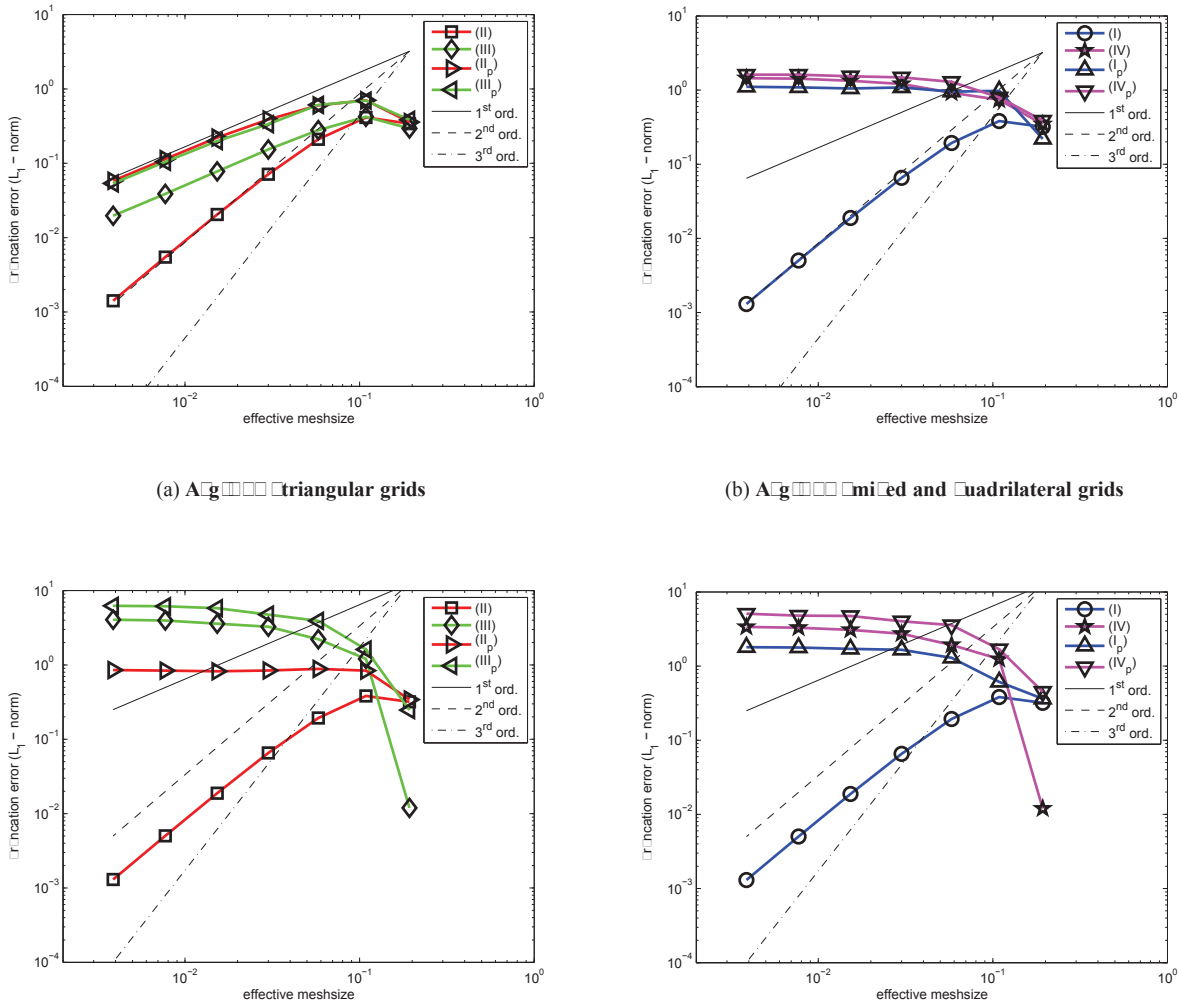
proportional to the aspect ratio.

B. Variation of gradient errors on grids of Class C

Table 3. Stencil for study of accuracy of gradient reconstruction on highly deformed grids.

Point	r	θ	x	y
0	R	0	0	0
N	$R + h_r$	0	0	h_r
S	$R - h_r$	0	0	$-h_r$
E	R	h_θ	$R\sin(h_\theta)$	$-R(1 - \cos(h_\theta))$
W	R	$-h_\theta$	$-R\sin(h_\theta)$	$-R(1 - \cos(h_\theta))$
NE	$R + h_r$	h_θ	$(R + h_r)\sin(h_\theta)$	$-(R + h_r)(1 - \cos(h_\theta))$
SW	$R - h_r$	$-h_\theta$	$-(R - h_r)\sin(h_\theta)$	$-(R - h_r)(1 - \cos(h_\theta))$

To illustrate the convergence property of gradient errors over a wide range of the deformation parameter Γ, a special computational test is designed. In the test, the gradient reconstruction is performed on a seven-point stencil corresponding to a Type (II) curved grid. The positions of stencil points (labeled in the compass notation) are shown in Table 3 in polar coordinates (r, θ) and in Cartesian coordinates (x, y) relative to the stencil center. In this test, radius $R = 1$ and radial mesh spacing $h_r = 2.5 \cdot 10^{-8}$ are kept fixed, the initial value of angular mesh spacing $h_\theta \approx 0.04$ is reduced by factor 2 in each of 13 refinement steps. With this "semi-refinement", Γ is reduced by factor 4 in each step, varying as $40,000 > \Gamma > 0.0005$ over the entire test. Figure 12 shows convergence of the Taylor expansion coefficients for the y-component of the gradient. The coefficients of terms that are not present in the figure are smaller than 10^{-10}. For the Taylor coefficients of the ULSQ y-gradient with a linear fit, a large magnitude and a flat convergence of the coefficient of U_{xx} observed in Figure 12(a) for $\Gamma \geq 1$ confirm an $O(1)$ accuracy of this gradient reconstruction method. In contrast, all Taylor coefficients of the ULSQ y-gradient with a quadratic fit shown



(a) A□g□□□ □triangular grids

(b) A□g□□□ □mi□ed and □uadrilateral grids

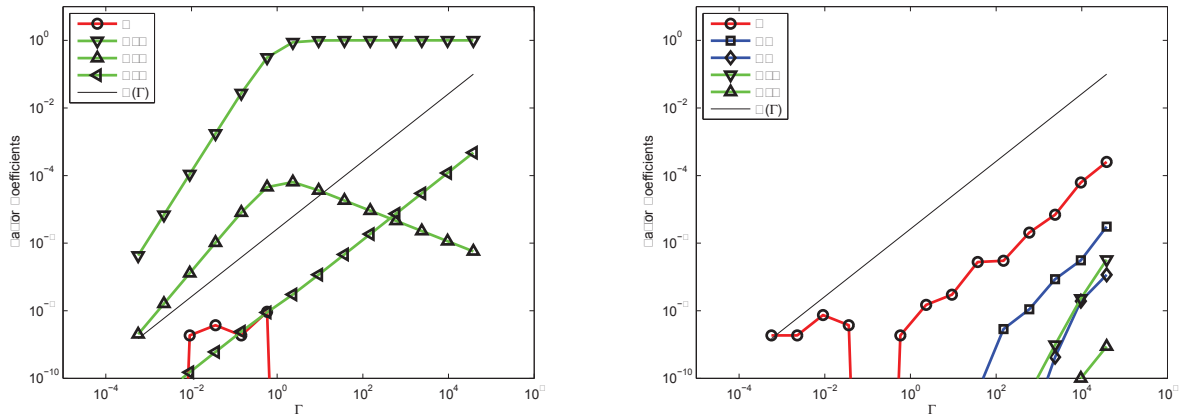
(c) GG□triangular grids

(d) GG□mi□ed and □uadrilateral grids

□igure □□. Viscous truncation errors on isotropic grids. □ anufactured solution is $U = \sin(\pi x - 2\pi y)$.

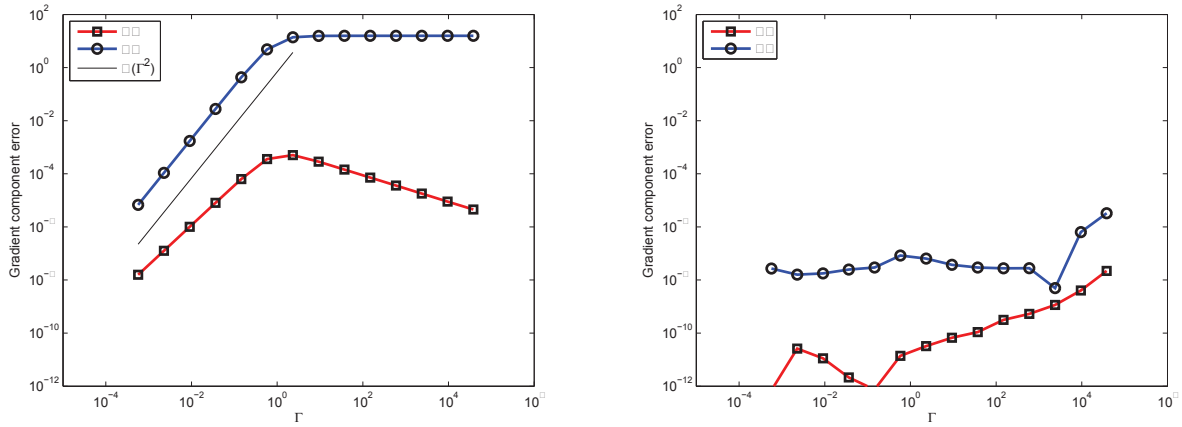
in Figure 1□(b) are small and converge with at least first order for high-Γ stencils.

The magnitudes of the relative errors for the GG scheme and for the ULSQ scheme with a quadratic fit are much smaller than the magnitude for the ULSQ scheme with a linear fit. Figure 16 shows the gradient errors measured at the center of the stencil for a radial solution $U = \sin(5\pi r)$. The gradient errors in Figure 16(a) confirm lack of accuracy for the ULSQ method with a linear fit on high-Γ grids. Low errors and □at convergence of the ULSQ method with a quadratic fit observed in Figure 16(b) are expected for accurate gradient reconstructions because the radial mesh size does not decrease in the test. This behavior indicates that for solutions varying predominantly in the radial direction, the gradient accuracy is determined by the radial mesh spacing and independent of Γ, which is a highly desirable property on high-Γ grids.



(a) □□□□ **linear □t** (b) □□□□ **□uadratic □t**

□igure □□. Con□ergence of □aylor coef□cients in semi□re□nement test.



(a) □□□□ **linear □t** (b) □□□□ **□uadratic □t**

□igure □□. Con□ergence of gradient errors in semi□re□nement test.

References

[1]Fidkowski, □. □ and □armofal, □. L., □Review of □utput-□ased Error Estimation and Grid Adaptation in □omputational Fluid □ynamics,□ *AIAA Journal*, □ol. 49, □o. 4, 2011, pp. 673□694.

[2]□ark, □., Lee-Rausch, E. □., and Rumsey, □. L., □FU□3□ and □FL3□ □omputations for the First □igh Lift □rediction Workshop,□AIAA □aper 2011-936, □anuary 2011.

[3]□erzins, □., □□esh Quality□A function of Geometry, Error Estimates, or □oth□□ *Engineering with Computers*, □ol. 1□, □o. 3, 1999, pp. 236□247.

[4]□nupp, □ □., □Algebraic □esh Quality □easures,□*SIAM J. Sci. Comp.*, □ol. 23, □o. 1, 2001, pp. 193□218.

[5] nupp, . ., Remarks on esh Quality, ral presentation, 4 th AIAA Aerospace Science eeting and Exhibit, Reno, , anuary 2007.

[6] astin, . W., Error induced by coordinate system, *Numerical Grid Generation*, edited by J. Thompson, North-Holand, New York, 1982, pp. 31–40.

[7] Lee, J. and Tsuei, Y., "A Formula for Estimation of Truncation Errors of Convection Terms in a Curvilinear Coordinate System," *J. Comput. Phys.*, Vol. 98, 1992, pp. 90–100.

[8] Diskin, B. and Thomas, J. L., "Notes on Accuracy of Finite Volume Discretization Schemes on Irregular Grids," *Appl. Num. Math.*, Vol. 60, No. 3, 2009, pp. 224–226, doi: 10.1016/j.apnum.2009.12.001.

[9] Forsyth, P. A. and Sammon, P. H., "Quadratic convergence for cell-centered grids," *Appl. Num. Math.*, Vol. 4, 1988, pp. 377–394.

[10] Kreiss, H. O., Manteuffel, T. A., Wendroff, B., and White, A. B., "Supra-convergence schemes on irregular grids," *Mathematics of Computations*, Vol. 47(176), October 1986, pp. 537–554.

[11] Manteuffel, T. A. and White, A. B., "The numerical solution of the second-order boundary value problem on nonuniform meshes," *Mathematics of Computations*, Vol. 47, No. 176, 1986, pp. 511–536.

[12] Babushka, I. and Aziz, A. K., "On the angle condition in the finite element method," *SIAM J. Numer. Anal.*, Vol. 13, April 1976, pp. 214–226.

[13] Krizek, M., "On the maximum angle condition for linear tetrahedral elements," *SIAM J. Numer. Anal.*, Vol. 29, No. 2, April 1992, pp. 513–520.

[14] Berzins, M., "Solution-Based Mesh Quality Indicators for Triangular and Tetrahedral Meshes," *International Journal of Computational Geometry and Applications*, Vol. 10, No. 3, 2000, pp. 333–346.

[15] Diskin, B., Thomas, J. L., Nielsen, E. J., Nishikawa, H., and White, J. A., "Comparison of Node-Centered and Cell-Centered Unstructured Finite-Volume Discretizations: Viscous Fluxes," *AIAA Journal*, Vol. 48, No. 7, 2010, pp. 1326–1338.

[16] Diskin, B. and Thomas, J. L., "Comparison of Node-Centered and Cell-Centered Unstructured Finite-Volume Discretizations: Inviscid Fluxes," *AIAA Journal*, Vol. 49, No. 4, 2011, pp. 836–854.

[17] Mavriplis, D. J., "Revisiting the Least-Square Procedure for Gradient Reconstruction on Unstructured Meshes," AIAA Paper 2003-3986, June 2003.

[18] Diskin, B. and Thomas, J. L., "Accuracy Analysis for Mixed-Element Finite-Volume Discretization Schemes," NIA Report 2007-08, August 2007.

[19] Thomas, J. L., Diskin, B., and Rumsey, C. L., "Towards Verification of Unstructured-Grid Solvers," *AIAA Journal*, Vol. 46, No. 12, 2008, pp. 3070–3079.

[20] Diskin, B. and Thomas, J. L., "Accuracy of Gradient Reconstruction on Grids with High Aspect Ratio," NIA Report 2008-12, December 2008.

[21] Haselbacher, A. C., *A Grid-Transparent Numerical Method for Compressible Viscous Flow on Mixed Unstructured Meshes*, Ph.D. thesis, Loughborough University, 1999.

[22] Anderson, W. K. and Bonhaus, D. L., "An implicit upwind algorithm for computing turbulent flows on unstructured grids," *Computers and Fluids*, Vol. 23, No. 1, 1994, pp. 1–21.

[23] Katz, A. and Sankaran, V., "Mesh quality effects on the accuracy of Euler and Navier-Stokes solutions on unstructured meshes," *Journal of Computational Physics*, Vol. 230, No. 20, 2011, pp. 7670–7686.

[24] Katz, A. and Sankaran, V., "An Efficient Correction Method to Obtain a Formally Third-Order Accurate Flow Solver for Node-Centered Unstructured Grids," *Journal on Scientific Computing*, 2011, doi: 10.1007/s10915-011-9515-1.

[25] Smith, T. M., Barone, M. F., Bond, R. B., Lorber, A. A., and Baur, D. G., "Comparison of Reconstruction Techniques for Unstructured Mesh Vertex Centered Finite Volume Scheme," AIAA Paper 2007-3958, June 2007.

[26] Roe, P. L., "Approximate Riemann solvers, parameter vectors, and difference schemes," *J. Comp. Phys.*, Vol. 43, No. 2, 1981, pp. 357–372.

[27] Aftosmis, M., Gaitonde, D., and Tavares, T. S., "Behavior of linear reconstruction techniques on unstructured meshes," *AIAA Journal*, Vol. 33, No. 11, 1995, pp. 2038–2049.

[28] Nishikawa, H., Diskin, B., and Thomas, J. L., "Critical study of agglomerated multigrid methods for diffusion," *AIAA Journal*, Vol. 48, No. 4, 2010, pp. 839–847.

[29] Thomas, J. L., Diskin, B., and Nishikawa, H., "A Critical Study of Agglomerated Multigrid Methods for Diffusion on Highly-Stretched Grids," *Computers and Fluids*, Vol. 41, No. 1, 2011, pp. 82–93.

[30] Trottenberg, U., Oosterlee, C. W., and Schüler, A., *Multigrid*, Academic Press, London, 2000.

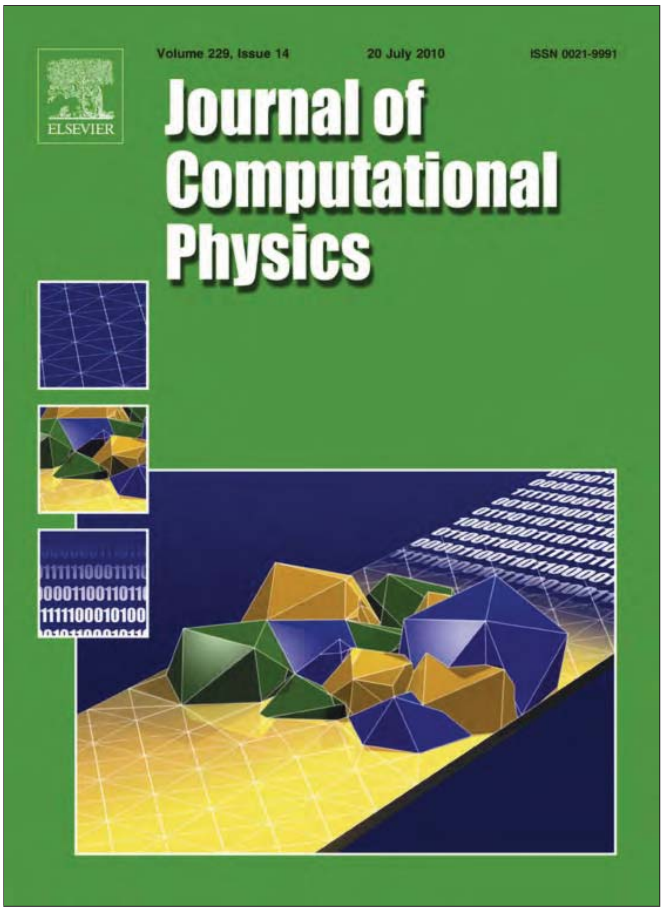
[31] Hirsch, C., *Numerical computation of internal and external flows. Vol.1, Fundamentals of numerical discretization*, A Wiley-Interscience publication, John Wiley & Sons, Inc., New York, NY, USA, 1988.

[32] Syrakos, A. and Goulas, A., "Estimate of the Truncation Error of Finite Volume Discretization of the Navier-Stokes Equations on Collocated Grids," *Int. J. Numer. Meth. Fluids*, Vol. 50, No. 1, 2006, pp. 103–130.

[33] Turkel, E., "Accuracy of Schemes with Nonuniform Meshes for Compressible Fluid Flows," *Applied Numerical Mathematics*, Vol. 2, No. 6, 1986, pp. 529–550.

[34] Giles, M. B., "Accuracy of node-based solutions on irregular meshes," *11th International Conference on Numerical Methods in Fluid Dynamics*, edited by D. L. Dwoyer, M. Y. Hussaini, and R. Voigt, Lecture Notes in Physics, v. 323, Springer-Verlag, 1989, pp. 369–373.

Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/copyright

Author's personal copy

Journal of Computational Physics 229 (2010) 5394–5407

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Local-in-time adjoint-based method for design optimization of unsteady flows

Nail K. Yamaleev [a,*], Boris Diskin [b,d], Eric J. Nielsen [c]

[a] Department of Mathematics, North Carolina A&T State University, Greensboro, NC 27411, USA
[b] National Institute of Aerospace, Hampton, VA 23666, USA
[c] Computational AeroSciences Branch, NASA Langley Research Center, Hampton, VA 23681, USA
[d] Department of Mechanical and Aerospace Engineering, University of Virginia, Charlottesville, VA 22904, USA

ARTICLE INFO

Article history:
Received 20 May 2009
Received in revised form 1 February 2010
Accepted 29 March 2010
Available online 4 April 2010

Keywords:
Time-dependent optimization
Discrete adjoint equations
Gradient methods
Design optimization
Euler equations

ABSTRACT

We present a new local-in-time discrete adjoint-based methodology for solving design optimization problems arising in unsteady aerodynamic applications. The new methodology circumvents storage requirements associated with the straightforward implementation of a global adjoint-based optimization method that stores the entire flow solution history for all time levels. This storage cost may quickly become prohibitive for large-scale applications. The key idea of the local-in-time method is to divide the entire time interval into several subintervals and to approximate the solution of the unsteady adjoint equations and the sensitivity derivative as a combination of the corresponding local quantities computed on each time subinterval. Since each subinterval contains relatively few time levels, the storage cost of the local-in-time method is much lower than that of the global methods, thus making the time-dependent adjoint optimization feasible for practical applications. Another attractive feature of the new technique is that the converged solution obtained with the local-in-time method is a local extremum of the original optimization problem. The new method carries no computational overhead as compared with the global implementation of adjoint-based methods. The paper presents a detailed comparison of the global- and local-in-time adjoint-based methods for design optimization problems governed by the unsteady compressible 2-D Euler equations.

© 2010 Elsevier Inc. All rights reserved.

1. Introduction

The continuous growth of computer power and the development of efficient and accurate computational tools now attract more attention to design optimization of unsteady flows. The time-dependent optimization problems arise in many aerodynamic applications including optimal design of helicopter rotors and turbomachinery blades, flutter and vibration control, noise reduction, active and passive flow control, etc. These problems can be formulated as minimization/maximization of appropriate cost functionals (e.g., lift, drag, torque, etc.) and can be solved by utilizing optimal control theory.

Among various optimization techniques available in the literature, adjoint-based gradient methods have recently grown in popularity, rapidly becoming one of the most widely used algorithms for solving a variety of steady and unsteady optimization problems. The adjoint methodology is particularly attractive for aerodynamic shape/design optimization problems that are characterized by the presence of a large number of design variables, yet relatively few constraints. In contrast to a classical forward mode differentiation approach whose computational cost is directly proportional to the number of design

* Corresponding author. Tel.: +1 336 285 2093; fax: +1 336 256 0876.
E-mail address: nkyamale@ncat.edu (N.K. Yamaleev).

0021-9991/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.03.045

Author's personal copy

variables, the adjoint methodology has the advantage of computing the cost functional gradients at a fixed expense independent of the number of design variables. Although the adjoint-based methods have been successfully used for problems of optimal design within the steady-state aerodynamics [1–4], applications of the adjoint formulation to time-dependent optimal design problems are still lacking. One of the main reasons why the time-dependent optimization has not been practically used in real-life aerodynamic applications is the storage cost involved. Straightforward global implementations of the discrete unsteady adjoint formulation require that the entire flow solution history should be available during the reverse time integration of the adjoint equations. For realistic 3-D design optimization problems, these storage requirements can quickly become prohibitive. For example, the storage cost of a typical discretization of the 3-D unsteady Reynolds Averaged Navier–Stokes (URANS) equations on a grid with 10^5 points per processor, which are integrated over 1000 time steps, is of the order of $O(10)$ Gb. Note that the storage cost may be significantly higher if a finer grid and more time levels are required to resolve the unsteady flow dynamics, and one stores not only the flow variables and grid coordinates, but also the grid velocities, face normals, control volumes, etc.

Several strategies aimed at circumventing these storage requirements have been developed and reported in the literature. All these methods can be divided into two groups. The first group of methods is "exact" in the sense that the primal and adjoint solutions computed using these methods exactly satisfy the corresponding equations of the original adjoint formulation. The most straightforward exact approach is to store the entire flow solution history to a hard disk (e.g., see [5–7]) and then use it during the reverse time integration of the adjoint equations. Note that for large-scale problems that are nonperiodic in time and require a very large number of time steps to integrate the governing equations, the storage and input/output costs may become prohibitively expensive. Another technique that provides a partial remedy to the storage problem is based on various checkpointing procedures which are performed either statically [8] or dynamically [9]. For this class of methods, the flow variables are stored only at so-called checkpoints whose number is much smaller than the total number of time steps required for integration of the primal and adjoint equations. During the backward-in-time integration of the adjoint equations, the required flow solution on each time subinterval between $(k-1)$th and kth checkpoints is recomputed by using the previously stored flow solution at the $(k-1)$th checkpoint as an initial condition. As a result, the flow solution should be stored only over a small time subinterval $[T_{k-1}, T_k]$ and at all checkpoints, thus significantly reducing the overall storage cost. However, as mentioned in [8,9], the computational cost increases by a factor of 2–3 because of the additional solves of the primal equations.

The key idea of the second group of methods is to reduce the storage cost by constructing sufficiently accurate approximations of either the original optimization problem or the corresponding governing equations. As a result, a solution obtained using these approximate techniques is suboptimal, i.e., not necessarily an extremum of the original time-dependent optimization problem. Among various suboptimal techniques, we would like to mention receding horizon control [10–12], system reduction [13–17], and nonlinear frequency domain methods [18,19]. The receding horizon techniques replace the original time-dependent optimization problem formulated on the entire time interval (the full time horizon) with a sequence of local optimal control problems defined on each time subinterval. Each of the subinterval problems, which are solved sequentially, consists of only a few (possibly one) time steps, so that its storage cost is much lower than that of the original unsteady optimization problem. This approach has been successfully used for optimal control problems governed by the 2-D incompressible Navier–Stokes equations. In [10], the receding horizon method is used for controlling the unsteady flow around a cylinder. Bewley et al. [11] use the receding horizon technique to re-laminarize the turbulent flow in a channel. In [12], Hou and Yan prove that the receding horizon method with distributed controls is stable for problems with a tracking-type functional governed by the 2-D incompressible Navier–Stokes equations. Note that the receding horizon techniques cannot be directly used for solving shape/design optimization problems. These methods compute only the local sensitivity derivative, while the global sensitivity derivative over the entire time interval of interest, which is required for solving the optimal design problems, is not available.

Another suboptimal approach that can significantly reduce the storage and computation costs is based on reduced-order or low-dimensional models of the original high-fidelity approximation of the Euler/Navier–Stokes equations. In [13], Tang et al. use a proper orthogonal decomposition (POD) reduced-order model based on a snapshot basis to control the unsteady wake flow around a cylinder. Hinze and Kunisch [14] present a POD-based boundary control technique that iteratively updates the low-order model and apply it to control the unsteady flow near a cylinder. In [15], two POD-based design optimization methods are used for inverse design of various airfoil shapes. POD modes and their Lagrangian sensitivities with respect to the shape variables are used to derive the POD basis to approximate a class of solutions over a range of design parameter values in [16]. This POD-based methodology is then applied to solving the two-dimensional flow past a square over a range of incidence angles. Modifications to the conventional POD procedure based on nonlinear projection for computing flow solutions are presented and demonstrated on several inverse design problems in [17]. Though POD-based reduced-order models can in principle drastically reduce the overall storage and CPU costs, their accuracy and consequently efficiency strongly depend on how well the POD basis represents the designed set of solutions. This problem associated with a proper selection of snapshots becomes a real challenge for essentially nonlinear compressible flows with shocks and contact discontinuities.

For periodic or quasi-periodic flows, the dimensionality of the corresponding unsteady discrete optimization problem can be reduced by expanding the flow solution in a Fourier series in time, thus reformulating the original optimization problem in the frequency domain. In [18], a gradient method based on the discrete adjoint equations and the corresponding boundary conditions in the frequency domain has been developed. This approach significantly reduces the storage and computation costs of the shape optimization of a 3-D wing oscillating at a constant frequency. An adjoint-based optimization procedure

Author's personal copy

based on the time-spectral formulation is developed and used for the analysis and shape design of helicopter rotors in forward flight in [19]. Similar to optimization techniques based on the POD reduced-order models, the time-spectral methods are suboptimal. Moreover, these methods are applicable only to time-periodic problems, and their efficiency strongly depends on the number of Fourier modes required to accurately approximate the solution of the unsteady governing equations.

In this paper, we present a new local-in-time discrete adjoint-based optimization methodology that combines the best features of both groups of methods outlined above. Similar to the suboptimal techniques, the new methodology tremendously reduces the overall storage cost by approximating the original adjoint equations on a set of local time subintervals, so that each subinterval involves only a few (possibly one) time steps. The distinctive features of the new local-in-time adjoint-based optimization algorithm are (1) the ability of the new method to converge to a local minimum of the original unsteady optimization problem; and (2) the fact that there is no additional computational overhead as compared with the global-in-time methods. Furthermore, since the global sensitivity derivative is evaluated at each optimization iteration of the new technique, it can be directly used for solving both optimal control and design optimization problems.

The rest of the paper is organized as follows. In Section 2, we present the discrete time-dependent optimization problem. Section 3 presents the conventional global-in-time adjoint-based method. The new local-in-time adjoint-based optimization method is introduced in Section 4. In Section 5, we validate the proposed time-dependent optimization methodology and evaluate its efficiency for three design optimization problems governed by the 2-D compressible Euler equations. We draw conclusions in Section 6.

2. Discrete design optimization problem

We consider a class of time-dependent design optimization problems governed by discretized unsteady flow equations written in the following form:

$$\frac{\mathbf{Q}^n - \mathbf{Q}^{n-1}}{\Delta t} + \mathbf{R}^n = \mathbf{0}, \tag{1}$$

where $\mathbf{Q} = \int_V \mathbf{U} dV$, $\mathbf{U}$ is a vector of the conserved variables, V is a control volume, $\mathbf{R}$ is the spatial undivided (by volume) flux residual, Δt is a time step, and superscript n denotes a time level number. The above discrete formulation (1) is very general and can be directly applied to the unsteady Euler or Reynolds-averaged Navier–Stokes equations [7]. In Eq. (1), the time derivative is approximated by using the implicit first-order backward-difference (BDF-1) formula; 2nd- and 3rd-order BDF formulae can also be used in the present formulation with minor modifications [7]. The governing (1) are discretized on a mesh which is given by the following equation:

$$\mathbf{G}(\mathbf{X}^n, \mathbf{D}) = \mathbf{0}, \tag{2}$$

where $\mathbf{X}^n$ is a mesh at time level n and $\mathbf{D}$ is a vector of the design variables. This time-dependent grid equation can easily adopt static, rigidly moving, and deforming meshes. For static grids considered in this paper, the grid $\mathbf{X}$ in Eq. (2) is independent of time, and the same grid equation is used for all time levels.

The discrete time-dependent optimization problem is formulated as follows:

$$\begin{cases} \min\limits_{\mathbf{D}\in\mathcal{D}_a} F_{\text{obj}}(\mathbf{D}), \quad F_{\text{obj}}(\mathbf{D}) = \sum\limits_{n=1}^{N} f^n(\mathbf{D}, \mathbf{Q}^n, \mathbf{X}^n)\Delta t, \\ \text{subject to Eqs.}(1) \text{ and } (2), \end{cases} \tag{3}$$

where $\mathbf{D}$ is a vector of the design variables, $\mathcal{D}_a$ is a set of admissible design parameters, which depends on specifics of the target physical system and ensures the existence of a solution of the optimization problem, N is the total number of time steps, $\mathbf{Q}$ is the solution of the unsteady flow Eq. (1), F_{obj} is an objective functional. The minimization problem (3) is very general and directly applicable to both active flow control and aerodynamic design optimization of unsteady flows.

To reduce the complexity of the optimization problem (3), without loss of generality, it is assumed that the objective functional F_{obj} is a scalar quantity. In the present analysis, f^n in Eq. (3) is defined as follows:

$$f^n = \sum_{j\in\Gamma_c} \left[C_j^n - \left(C_j^{\text{target}} \right)^n \right]^2, \tag{4}$$

where C_j is an aerodynamic quantity such as the lift or the pressure coefficient on a controlled boundary surface Γ_c, C_j^{target} is a given target value of C_j. Thus, F_{obj} given by Eqs. (3) and (4) is a matching-type functional.

3. Global-in-time adjoint-based optimization method

The discrete time-dependent optimization problem (3) is solved by the method of Lagrange multipliers which is used to enforce the governing Eq. (1) as constraints. The discrete Lagrangian functional is defined as follows:

$$L(\mathbf{D}, \mathbf{Q}, \mathbf{X}, \mathbf{\Lambda}_f, \mathbf{\Lambda}_g) = \sum_{n=0}^{N} f^n \Delta t + \sum_{n=1}^{N} \left[\mathbf{\Lambda}_f^n \right]^T \left(\frac{\mathbf{Q}^n - \mathbf{Q}^{n-1}}{\Delta t} + \mathbf{R}^n \right) \Delta t + \left[\mathbf{\Lambda}_f^0 \right]^T (\mathbf{Q}^0 - \mathbf{Q}^{\text{in}}) + \sum_{n=0}^{N} \left[\mathbf{\Lambda}_g^n \right]^T \mathbf{G}^n \Delta t, \tag{5}$$

Author's personal copy

where $\mathbf{\Lambda}_f$ and $\mathbf{\Lambda}_g$ are flow and grid Lagrange multipliers (adjoint variables), respectively, time levels $n = 0$ and $n = N$ correspond to times $t = 0$ and $t = T_{\text{final}}$, $\mathbf{Q}^{\text{in}}$ is an initial condition for the flow (1), f^n is given by Eq. (4), and $\mathbf{R}^n = \mathbf{R}(\mathbf{Q}^n, \mathbf{X}^n, \mathbf{D})$ is the spatial undivided residual.

The sensitivity derivative is obtained by differentiating the Lagrangian with respect to $\mathbf{D}$, which yields

$$\begin{aligned}\frac{dL}{d\mathbf{D}} = &\sum_{n=0}^{N}\frac{\partial f^n}{\partial \mathbf{D}}\Delta t + \sum_{n=1}^{N}[\mathbf{\Lambda}_f^n]^T\frac{\partial \mathbf{R}^n}{\partial \mathbf{D}}\Delta t + \sum_{n=1}^{N}\left(\frac{[\mathbf{\Lambda}_f^n]^T - [\mathbf{\Lambda}_f^{n+1}]^T}{\Delta t} + [\mathbf{\Lambda}_f^n]^T\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}^n} + \frac{\partial f^n}{\partial \mathbf{Q}^n}\right)\frac{\partial \mathbf{Q}^n}{\partial \mathbf{D}}\Delta t \\ &+ \left(\frac{\left[\mathbf{\Lambda}_f^0\right]^T - \left[\mathbf{\Lambda}_f^1\right]^T}{\Delta t} + \frac{\partial f^0}{\partial \mathbf{Q}^0}\right)\frac{\partial \mathbf{Q}^0}{\partial \mathbf{D}}\Delta t - \left[\mathbf{\Lambda}_f^0\right]^T\frac{\partial \mathbf{Q}^{\text{in}}}{\partial \mathbf{D}} + \sum_{n=1}^{N}\left(\frac{\partial f^n}{\partial \mathbf{X}^n} + \left[\mathbf{\Lambda}_f^n\right]^T\frac{\partial \mathbf{R}^n}{\partial \mathbf{X}^n} + \left[\mathbf{\Lambda}_g^n\right]^T\frac{\partial \mathbf{G}^n}{\partial \mathbf{X}^n}\right)\frac{\partial \mathbf{X}^n}{\partial \mathbf{D}}\Delta t \\ &+ \sum_{n=0}^{N}[\mathbf{\Lambda}_g^n]^T\frac{\partial \mathbf{G}^n}{\partial \mathbf{D}}\Delta t + \left(\frac{\partial f^0}{\partial \mathbf{X}^0} + \left[\mathbf{\Lambda}_g^0\right]^T\frac{\partial \mathbf{G}^0}{\partial \mathbf{X}^0}\right)\frac{\partial \mathbf{X}^0}{\partial \mathbf{D}}\Delta t, \end{aligned} \tag{6}$$

where $\mathcal{A}^{N+1} = 0$. In the above equation and throughout the paper, we use the following notations. The derivative of a scalar $c \in R$ with respect to a column vector $\mathbf{a} \in R^m$, $\partial c/\partial \mathbf{a}$, is the row vector: $\left[\frac{\partial c}{\partial a_1}, \ldots, \frac{\partial c}{\partial a_m}\right]$, and the derivative of a column vector $\mathbf{b} \in R^l$ with respect to a column vector $\mathbf{a} \in R^m$ is the $l \times m$ matrix:

$$\frac{\partial \mathbf{b}}{\partial \mathbf{a}} = \begin{bmatrix} \frac{\partial b_1}{\partial a_1} & \cdots & \frac{\partial b_1}{\partial a_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial b_l}{\partial a_1} & \cdots & \frac{\partial b_l}{\partial a_m} \end{bmatrix}.$$

For aerodynamic design optimization problems, the number of design variables is typically very large. Therefore, the computation of $\partial \mathbf{Q}^n/\partial \mathbf{D}$ and $\partial \mathbf{X}^n/\partial \mathbf{D}$ is extremely expensive in terms of the CPU time, because it requires as many solves of the flow and grid equations as the total number of the design variables involved. To eliminate the $\partial \mathbf{Q}^n/\partial \mathbf{D}$ and $\partial \mathbf{X}^n/\partial \mathbf{D}$ terms from the sensitivity derivative, their coefficients on the right-hand side of Eq. (6) are set equal to zero, thus leading to the following adjoint equations for determining the flow adjoint variables:

$$\begin{cases} \frac{1}{\Delta t}\mathbf{\Lambda}_f^N + \left[\frac{\partial \mathbf{R}^N}{\partial \mathbf{Q}^N}\right]^T\mathbf{\Lambda}_f^N = -\left[\frac{\partial f^N}{\partial \mathbf{Q}^N}\right]^T, & \text{for } n = N, \\ \frac{1}{\Delta t}\left(\mathbf{\Lambda}_f^n - \mathbf{\Lambda}_f^{n+1}\right) + \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}^n}\right]^T\mathbf{\Lambda}_f^n = -\left[\frac{\partial f^n}{\partial \mathbf{Q}^n}\right]^T, & \text{for } 2 \leqslant n \leqslant N-1, \\ \frac{1}{\Delta t}\left(\mathbf{\Lambda}_f^0 - \mathbf{\Lambda}_f^1\right) = -\left[\frac{\partial f^0}{\partial \mathbf{Q}^0}\right]^T, & \text{for } n = 1, \end{cases} \tag{7}$$

and the grid adjoint variables:

$$\begin{cases} \left[\frac{\partial \mathbf{G}^n}{\partial \mathbf{X}^n}\right]^T\mathbf{\Lambda}_g^n = -\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{X}^n}\right]^T\mathbf{\Lambda}_f^n - \left[\frac{\partial f^n}{\partial \mathbf{X}^n}\right]^T, & \text{for } 1 \leqslant n \leqslant N, \\ \left[\frac{\partial \mathbf{G}^0}{\partial \mathbf{X}^0}\right]^T\mathbf{\Lambda}_g^0 = -\left[\frac{\partial f^0}{\partial \mathbf{X}^0}\right]^T, & \text{for } n = 0. \end{cases} \tag{8}$$

The main advantage of the adjoint formulation is that at each optimization iteration, the adjoint Eqs. (7) and (8) are independent of $\mathbf{D}$ and should be solved once regardless of the number of the design variables. Equations (7) and (8) represent linear systems of equations for the flow and grid adjoint variables, respectively. The flow adjoint equations do not depend on $\mathbf{\Lambda}_g$. Therefore, the systems of Eqs. (7) and (8) are weakly coupled and can be solved sequentially. Once the solution of the flow adjoint equations at the nth time level is available, then $\mathbf{\Lambda}_f^n$ is substituted into Eq. (8) which is solved to determine the grid adjoint variables $\mathbf{\Lambda}_g^n$ at the same time level.

In contrast to the primal flow Eq. (1), the first term in each Eq. (7) approximates the negative time derivative, thus indicating that the unsteady flow adjoint equations have to be integrated backward in time. Therefore, the flow solution $\mathbf{Q}^n$, which is used for computing the matrix $\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}^n}\right]^T$ and the vector $\left[\frac{\partial f^n}{\partial \mathbf{Q}^n}\right]^T$ in Eq. (7), must be available for all time levels during the backward-in-time integration of the flow adjoint equations. For the global time-dependent adjoint-based method, the entire flow solution history for all time levels is stored during the forward sweep in time. As a result, the storage cost of the time-dependent adjoint formulation is much higher than that of the steady state adjoint formulation.

With the flow and grid adjoint variables found from Eqs. (7) and (8), the sensitivity derivative is calculated as follows:

$$\frac{dL}{d\mathbf{D}} = \sum_{n=0}^{N}\frac{\partial f^n}{\partial \mathbf{D}}\Delta t + \sum_{n=1}^{N}\left[\mathbf{\Lambda}_f^n\right]^T\frac{\partial \mathbf{R}^n}{\partial \mathbf{D}}\Delta t + \sum_{n=0}^{N}\left[\mathbf{\Lambda}_g^n\right]^T\frac{\partial \mathbf{G}^n}{\partial \mathbf{D}}\Delta t - \left[\mathbf{\Lambda}_f^0\right]^T\frac{\partial \mathbf{Q}^{\text{in}}}{\partial \mathbf{D}}. \tag{9}$$

A minimum of the functional given by Eq. (5) is found by the steepest descent method in which each step of the optimization cycle is taken in the negative gradient direction

Author's personal copy

$$\mathbf{D}_{i+1} = \mathbf{D}_i - \delta_i \left[\frac{dL}{d\mathbf{D}}\right]_i^T, \tag{10}$$

where δ_i is an optimization step size which is chosen adaptively [22], i is a steepest descent iteration number, $\mathbf{D}$ is a vector of the design variables. The sensitivity derivative $dL/d\mathbf{D}$ in Eq. (10) is computed using Eq. (9) which requires the solution of the flow and grid adjoint Eqs. (7) and (8). When the flow and grid adjoint equations are integrated backward in time, the sensitivity derivative at each time step is computed and added to its value at the previous time step. At $n = 0$, the complete sensitivity derivative vector is available and used in Eq. (10) for updating the vector of design variable $\mathbf{D}_{i+1}$. Then, the entire optimization cycle is repeated until either $|F_{\text{obj}}^{i+1} - F_{\text{obj}}^{i}| < \epsilon_1$ or $\|dL/\mathbf{D}_i\| < \epsilon_2$, where ϵ_1 and ϵ_2 are given tolerances and $\|\cdot\|$is an appropriate norm. The above procedure can be summarized in the form of the following global-in-time (GT) adjoint-based algorithm:

Algorithm 1. Global-in-time (GT) adjoint-based method

(1) Choose $\mathbf{D}_1$ and set $i = 1$.
(2) Solve Eq. (1) forward in time for $\mathbf{Q}^0, \ldots, \mathbf{Q}^N$ and store $\mathbf{Q}^n$, $1 \leqslant n \leqslant N$.
(3) Solve Eqs. (7) and (8) backward in time for Λ_f^n and Λ_g^n, $1 \leqslant n \leqslant N$.
(4) Evaluate $\frac{dL}{d\mathbf{D}}$ using Eq. (9).
(5) Choose δ_i and update $\mathbf{D}_{i+1}$ using Eq. (10).
(6) If $|F_{\text{obj}}^{i+1} - F_{\text{obj}}^{i}| > \epsilon_1$ and $\|dL/\mathbf{D}_i\| > \epsilon_2$, set $i = i + 1$ and go to step 2; otherwise stop.

This GT algorithm possesses the following property. Namely, if the objective functional is defined to be zero on the entire time interval of interest except the final time level, i.e., $F_{\text{obj}} = f^N \Delta t$, then the corresponding flow and grid adjoint variables exponentially decay to zero in reverse time. This property is a direct consequence of a similarity between the homogeneous flow adjoint Eq. (7) and error equations. Indeed, assuming that $\mathbf{Q}$ is the exact solution of the semi-discrete flow equations $\mathbf{Q}_t + \mathbf{R}(\mathbf{Q}) = 0$ and $\mathbf{e}$ is a solution error caused by a small perturbation of the initial condition, we have

$$\frac{\partial(\mathbf{Q} + \mathbf{e})}{\partial t} + \mathbf{R}(\mathbf{Q} + \mathbf{e}) = 0. \tag{11}$$

Linearizing the above equation with respect to $\mathbf{Q}$ yields

$$\frac{\partial \mathbf{e}}{\partial t} + \frac{\partial \mathbf{R}}{\partial \mathbf{Q}}\mathbf{e} = 0. \tag{12}$$

The homogeneous flow adjoint equations obtained from Eq. (7) by setting $\partial f^n/\partial \mathbf{Q}^n = 0$ for all $n \leqslant N - 1$ are similar to a first-order approximation of the transposed error equation (12). The linear Eqs. (7) and (12) can be integrated in time, thus leading to the following matrix exponential solutions: $\exp(-[\partial \mathbf{R}/\partial \mathbf{Q}]t)\mathbf{e}_0$ and $\exp(-[\partial \mathbf{R}/\partial \mathbf{Q}]^T(T_{\text{final}} - t))\Lambda_f^N$ for the error and flow adjoint vectors, respectively. For strongly stable numerical schemes, all eigenvalues of the Jacobian matrix $-\partial \mathbf{R}/\partial \mathbf{Q}$ and its transpose $-[\partial \mathbf{R}/\partial \mathbf{Q}]^T$ are located in the left half of the complex plane. Therefore, the numerical error and the flow adjoints exponentially decay in forward and reverse times, respectively. For the flow adjoints, the decay is expected to be strong, because the time derivative of the flow adjoint vector approaches zero at $t \to 0$, as follows from the last equation in Eq. (7) with $\partial f^0/\partial \mathbf{Q}^0 = 0$. From Eq. (8) with $\partial f^n/\partial \mathbf{X}^n = 0$ for $1 \leqslant n \leqslant N - 1$ it follows that the exponential decay of the flow adjoint vector to zero in reverse time results in a similar decay of the grid adjoint vector. Thus, the major contributions of the flow and grid adjoints to the sensitivity derivative come from the final time levels, dominating contributions from intermediate and initial time levels. Numerical results corroborating the above estimates are presented in Section 5.

4. Local-in-time adjoint-based optimization method

As has been mentioned in the foregoing section, at each iteration of the GT method, the flow equations are integrated forward in time while the adjoint equations are integrated backward in time over the entire time interval considered. Since the adjoint operators in Eqs. (7) and (8) depend on $\mathbf{Q}^n$ and $\mathbf{X}^n$, the solution of the flow problem and the corresponding computational grid in the GT algorithm are stored for all time levels over which the optimization problem is solved. For realistic 3-D optimization problems, these storage requirements can quickly become prohibitive. This motivates us to consider local-in-time strategies to reduce the storage cost of the GT method presented in Section 3.

We begin by dividing the entire time interval into K subintervals such that $0 = T_0 < \cdots < T_K = N\Delta t = T_{\text{final}}$, where $T_k = \Delta t\, N_k$, $K \leqslant N$, and Δt is a constant time step used for integrating the primal and adjoint equations. In general, this partitioning can be chosen so that each subinterval contains one or several time steps of the time-marching scheme used for solving the governing equations. The main idea of the proposed strategy is based on the observation that the global sensitivity derivative given by Eq. (9) can be represented as a sum of local sensitivity derivatives defined on each time subinterval. That is

$$\frac{dL}{d\mathbf{D}} = \sum_{k=1}^{K} \frac{dL^k}{d\mathbf{D}}, \tag{13}$$

Author's personal copy

where the local Lagrangian functionals are given by

$$L^k = \begin{cases} \sum\limits_{n=N_{k-1}+1}^{N_k} f^n \Delta t + \sum\limits_{n=N_{k-1}+1}^{N_k} \left[\Lambda_f^n\right]^T \left(\frac{\mathbf{Q}^n - \mathbf{Q}^{n-1}}{\Delta t} + \mathbf{R}^n\right)\Delta t, & \text{for } 2 \leqslant k \leqslant K, \\ \sum\limits_{n=0}^{N_1} f^n \Delta t + \sum\limits_{n=1}^{N_1} \left[\Lambda_f^n\right]^T \left(\frac{\mathbf{Q}^n - \mathbf{Q}^{n-1}}{\Delta t} + \mathbf{R}^n\right)\Delta t + \left[\Lambda_f^0\right]^T \left(\mathbf{Q}^0 - \mathbf{Q}^{\text{in}}\right), & \text{for } k = 1. \end{cases} \tag{14}$$

In this section, without loss of generality, the grid terms are omitted for simplicity.

The adjoint equations corresponding to the local Lagrangian functionals, L^k for $1 \leqslant k \leqslant K$, can be derived by using the same adjoint-based approach described in the foregoing section. Differentiating each local Lagrangian, L^k, with respect to $\mathbf{D}$ and taking into account the contribution from L^{k+1} yields the following flow adjoint equations on subinterval $(T_{k-1}, T_k]$:

$$\begin{cases} \frac{1}{\Delta t}\left(\Lambda_f^{N_k} - \Lambda_f^{N_k+1}\right) + \left[\frac{\partial \mathbf{R}^{N_k}}{\partial \mathbf{Q}^{N_k}}\right]^T \Lambda_f^{N_k} = -\left[\frac{\partial f^{N_k}}{\partial \mathbf{Q}^{N_k}}\right]^T, & \text{for } n = N_k, \\ \frac{1}{\Delta t}\left(\Lambda_f^{n} - \Lambda_f^{n+1}\right) + \left[\frac{\partial \mathbf{R}^{n}}{\partial \mathbf{Q}^{n}}\right]^T \Lambda_f^{n} = -\left[\frac{\partial f^{n}}{\partial \mathbf{Q}^{n}}\right]^T, & \text{for } N_{k-1}+1 \leqslant n \leqslant N_k - 1, \\ \frac{1}{\Delta t}\left(\Lambda_f^{0} - \Lambda_f^{1}\right) = -\left[\frac{\partial f^{0}}{\partial \mathbf{Q}^{0}}\right]^T, & \text{for } n = 0, \end{cases} \tag{15}$$

where Λ_f^n is the solution of the flow adjoint equations defined for $N_{k-1} < n \leqslant N_k$. The presence of the $\Lambda_f^{N_k+1}$ term in Eq. (15) indicates that the system of adjoint equations on subinterval $(T_{k-1}, T_k]$ is coupled with the system of adjoint equations defined on the next subinterval $(T_k, T_{k+1}]$. In fact, Eq. (15) for $1 \leqslant k \leqslant K$ represents a set of coupled systems of adjoint equations on the entire time interval $[0, T_{\text{final}}]$, which is equivalent to the original adjoint Eq. (5). As a result, the flow solution for all time levels has to be available when these adjoint Eq. (15) for $1 \leqslant k \leqslant K$ are integrated backward in time.

To reduce the storage cost, we decouple the set of (15) for $1 \leqslant k \leqslant K$ by approximating $\Lambda_f^{N_k+1}$ as $\widetilde{\Lambda}_f$, thus leading to the following local-in-time adjoint equations defined on $(T_{k-1}, T_k]$:

$$\begin{cases} \frac{1}{\Delta t}\left(\widehat{\Lambda}_f^{N_k} - \widetilde{\Lambda}_f\right) + \left[\frac{\partial \mathbf{R}^{N_k}}{\partial \mathbf{Q}^{N_k}}\right]^T \widehat{\Lambda}_f^{N_k} = -\left[\frac{\partial f^{N_k}}{\partial \mathbf{Q}^{N_k}}\right]^T, & \text{for } n = N_k, \\ \frac{1}{\Delta t}\left(\widehat{\Lambda}_f^{n} - \widehat{\Lambda}_f^{n+1}\right) + \left[\frac{\partial \mathbf{R}^{n}}{\partial \mathbf{Q}^{n}}\right]^T \widehat{\Lambda}_f^{n} = -\left[\frac{\partial f^{n}}{\partial \mathbf{Q}^{n}}\right]^T, & \text{for } N_{k-1}+1 \leqslant n \leqslant N_k - 1, \\ \frac{1}{\Delta t}\left(\widehat{\Lambda}_f^{0} - \widehat{\Lambda}_f^{1}\right) = -\left[\frac{\partial f^{0}}{\partial \mathbf{Q}^{0}}\right]^T, & \text{for } n = 0, \end{cases} \tag{16}$$

where $\widehat{\Lambda}_f^n$ is an approximation of the corresponding adjoint solution Λ_f^n. The last equation in Eq. (16) is used only on the first subinterval $[T_0, T_1]$ corresponding to $k = 1$.

It should be noted that the partitioning of the entire time interval into subintervals does not alter the solution of the flow equations. Indeed, the flow equations are integrated forward in time beginning from $n = 0$ that corresponds to the initial condition of the original flow problem. The flow solution obtained at the end of the first time subinterval, $\mathbf{Q}^{N_1}$, is used as an initial condition for the second subinterval, and so on. In the case that a second- or higher order backward difference (BDF) scheme is employed for discretization of the time derivative, flow solutions at the corresponding number (depending on the BDF scheme used) of time levels of the previous subinterval are employed to continue the integration of the governing equations on the current time subinterval. The result is that the flow solution obtained in this manner is identical to that computed on the entire time interval $[0, T_{\text{final}}]$ by using a single sweep in time.

With the local flow adjoint variables $\widehat{\Lambda}_f^n$ satisfying Eq. (16), the local sensitivity derivative on each subinterval $(T_{k-1}, T_k]$ is calculated as follows:

$$\frac{d\widehat{L}^k}{d\widehat{\mathbf{D}}} = \begin{cases} \sum\limits_{n=N_{k-1}+1}^{N_k} \frac{\partial f^n}{\partial \widehat{\mathbf{D}}} \Delta t + \sum\limits_{n=N_{k-1}+1}^{N_k} \left[\widehat{\Lambda}_f^n\right]^T \frac{\partial \mathbf{R}^n}{\partial \widehat{\mathbf{D}}} \Delta t, & \text{for } 2 \leqslant k \leqslant K, \\ \sum\limits_{n=0}^{N_1} \frac{\partial f^n}{\partial \widehat{\mathbf{D}}} \Delta t + \sum\limits_{n=1}^{N_1} \left[\widehat{\Lambda}_f^n\right]^T \frac{\partial \mathbf{R}^n}{\partial \widehat{\mathbf{D}}} \Delta t - \left[\widehat{\Lambda}_f^0\right]^T \frac{\partial \mathbf{Q}^{\text{in}}}{\partial \widehat{\mathbf{D}}}, & \text{for } k = 1. \end{cases} \tag{17}$$

By analogy with Eq. (13), the approximate global sensitivity derivative, $\frac{d\widehat{L}}{d\widehat{\mathbf{D}}}$, is computed as

$$\frac{d\widehat{L}}{d\widehat{\mathbf{D}}} = \sum_{k=1}^{K} \frac{d\widehat{L}^k}{d\widehat{\mathbf{D}}}. \tag{18}$$

Once the global sensitivity derivative is available at the last Kth time subinterval, the vector of design variables is updated by using the steepest descent method

$$\widehat{\mathbf{D}}_{i+1} = \widehat{\mathbf{D}}_i - \delta_i \left[\frac{d\widehat{L}}{d\widehat{\mathbf{D}}}\right]_i^T. \tag{19}$$

Similar to the GT method, the steepest descent iterations are repeated until either $\left|F_{\text{obj}}^{i+1} - F_{\text{obj}}^{i}\right| < \epsilon_1$ or $\|d\widehat{L}/\widehat{\mathbf{D}}_i\| < \epsilon_2$, where ϵ_1 and ϵ_2 are user-specified tolerances and $\|\cdot\|$is an appropriate norm.

Author's personal copy

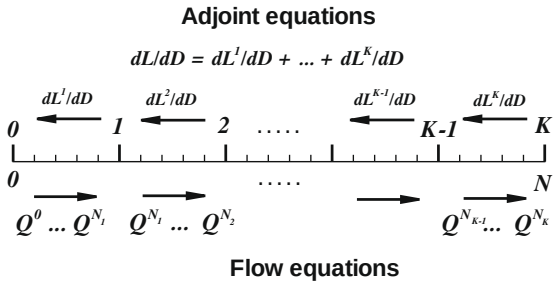


Fig. 1. A sketch of the local-in-time adjoint-based algorithm.

Comparing Eqs. (15) and (16), the following observation can be made. If $\widetilde{\mathbf{\Lambda}}_f$ in Eq. (16) is set equal to zero, then each local system of adjoint Eq. (16) defined on a given time subinterval $(T_{k-1}, T_k]$ is independent of the other adjoint equations defined on $[0, T_{k-1}] \cup (T_k, T_{\text{final}}]$. Thus, the local systems of adjoint Eq. (16) can be solved sequentially starting from the first time subinterval ($k = 1$) and marching *forward* one subinterval by another up to $k = K$. Within each subinterval $(T_{k-1}, T_k]$, the local adjoint equations (16) are integrated backward in time. Although the systems of local adjoint equations (16) defined on k and $k + 1$ time subintervals are decoupled if $\widetilde{\mathbf{\Lambda}}_f = 0$, they cannot be solved simultaneously because each system of adjoint equations requires the flow solutions to be available on the same time subinterval. The local sensitivity derivatives calculated on each subinterval using Eq. (17) are then summed up to give the global sensitivity derivative on the entire time interval $[0, T_{\text{final}}]$, as shown in Fig. 1. Note that the flow adjoints obtained with the local Eq. (16) for $1 \leqslant k \leqslant K$ and the corresponding approximate total sensitivity derivative given by Eq. (18) are, in principle, not equal to those given by Eqs. (7) and (9), i.e., $\widehat{\mathbf{\Lambda}}_f \neq \mathbf{\Lambda}_f$ and $d\widehat{L}/d\widehat{\mathbf{D}} \neq dL/d\mathbf{D}$ on $(T_{k-1}, T_k]$, where $\mathbf{\Lambda}_f$ denotes the solution of the global flow adjoint Eq. (7). Though $d\widehat{L}/d\widehat{\mathbf{D}}$ is only an approximation to $dL/d\mathbf{D}$ given by Eq. (9), this approach reduces the storage cost by a factor of K as compared with the GT algorithm. Indeed, since the local adjoint equations on each time subinterval $(T_k, T_{k+1}]$ can be solved independently of the adjoint equations defined on the other subintervals, only the flow solutions for the current subinterval, $\mathbf{Q}^{N_{k-1}+1}, \ldots, \mathbf{Q}^{N_k}$, have to be stored, thus drastically reducing the storage cost. Further in the paper, this algorithm with $\widetilde{\mathbf{\Lambda}}_f = \mathbf{0}$ is referred as a simplified local-in-time (SLT) method.

Another observation based on the comparative analysis of Eqs. (7), (9) and (16), (17) is that the entire set of systems of local adjoint Eq. (16) for $1 \leqslant k \leqslant K$ is identical to the global adjoint Eq. (15) and consequently to Eq. (7), if $\widetilde{\mathbf{\Lambda}}_f$ in Eq. (16) is set to be $\mathbf{\Lambda}_f^{N_k+1}$. In spite of the fact that this approach provides complete consistency of the local and global adjoint equations, it destroys the locality of the adjoint Eq. (16) and therefore requires the same full storage as the GT method.

These considerations suggest that $\widetilde{\mathbf{\Lambda}}_f$ in Eq. (16) should be chosen such that it preserves the locality of each system of adjoint equations defined on subinterval $(T_{k-1}, T_k]$ and provides a good approximation of $\mathbf{\Lambda}_f^{N_k+1}$. To satisfy these constraints, we propose to choose $\widetilde{\mathbf{\Lambda}}_f$ as

$$(\widetilde{\mathbf{\Lambda}}_f)_i = \left(\mathbf{\Lambda}_f^{N_k+1}\right)_{i-1}, \tag{20}$$

where i is a design iteration number. In other words, the required vector of adjoint variables at time level $N_k + 1$ is taken from the previous iteration of the steepest descent method (19). This local-in-time (LT) adjoint-based strategy for solving the minimization problem (3) and (4) is summarized in the form of the following algorithm:

Algorithm 2. Local-in-time (LT) adjoint-based method

(1) Choose $\widehat{\mathbf{D}}_1$, and K; set $k = 1$, $i = 1$, $\left(\widehat{\mathbf{\Lambda}}_f^{N_k+1}\right)_0 = 0$ for $1 \leqslant k \leqslant K$, and $\frac{d\widehat{L}}{d\widehat{\mathbf{D}}} = 0$.
(2) Solve Eq. (1) for $\mathbf{Q}^{N_{k-1}+1}, \ldots, \mathbf{Q}^{N_k}$ forward in time on $(T_{k-1}, T_k]$; store $\mathbf{Q}^n$ for $N_{k-1} + 1 \leqslant n \leqslant N_k$.
(3) If $i \leqslant i_s$ set $\widetilde{\mathbf{\Lambda}}_f = \mathbf{0}$, otherwise $\widetilde{\mathbf{\Lambda}}_f = \left(\widehat{\mathbf{\Lambda}}_f^{N_k+1}\right)_{i-1}$, where i_s is a user-defined number of iterations.
(4) Solve Eq. (16) backward in time for $\widehat{\mathbf{\Lambda}}_f^{N_{k-1}+1}, \ldots, \widehat{\mathbf{\Lambda}}_f^{N_k}$; store $\widehat{\mathbf{\Lambda}}_f^{N_{k-1}+1}$.
(5) Evaluate $\frac{d\widehat{L}^k}{d\widehat{\mathbf{D}}}$ by using Eq. (17).
(6) Set $\frac{d\widehat{L}}{d\widehat{\mathbf{D}}} = \frac{d\widehat{L}}{d\widehat{\mathbf{D}}} + \frac{d\widehat{L}^k}{d\widehat{\mathbf{D}}}$.
(7) Set $k = k + 1$, if $k \leqslant K$ go to step 2; otherwise continue.
(8) Calculate $\widehat{\mathbf{D}}_{i+1}$ using Eq. (19).
(9) If $|F_{\text{obj}}^{i+1} - F_{\text{obj}}^{i}| > \epsilon_1$ and $\|d\widehat{L}/d\widehat{\mathbf{D}}_i\| > \epsilon_2$ set $k = 1$, $i = i + 1$, $\frac{d\widehat{L}}{d\widehat{\mathbf{D}}} = 0$ and go to step 2; otherwise stop.

The above choice of $\widetilde{\mathbf{\Lambda}}_f$ given by Eq. (20) significantly reduces the storage cost as compared to the GT method. Indeed, for the LT method, the flow solution should be stored only at those time levels that belong to the current time subinterval

Author's personal copy

$(T_{k-1}, T_k]$. In addition, flow adjoint solutions at $K-1$ time levels from the previous optimization cycle, $\left(\widehat{\Lambda}_f^{N_k+1}\right)_{i-1}$ for $1 \leqslant k \leqslant K-1$, should also be stored, as follows from Eq. (20). Therefore, the overall storage cost of the LT algorithm is $O(K+N/K)$ flow variables versus $O(N)$ flow variables required for the GT method. Since $K+N/K$ achieves its minimum value at $K=\sqrt{N}$, the storage cost of the LT algorithm can be minimized if the number of time subintervals K is set equal to $\sqrt{N}$, where N is the total number of time levels. For $K=\sqrt{N}$, the total storage cost of the LT algorithm is $\sqrt{N}/2$ times less than that of the GT algorithm. The savings are even more significant when dynamic grids are involved.

In addition to the significant storage savings, another key advantage of the LT algorithm is that upon convergence, the set of local-in-time adjoint equations becomes identical to the original adjoint Eq. (7), thus providing full consistency between the local and global methods. In other words, the converged solution obtained with LT method is a local minimum of the original optimization problem (3). Indeed, assuming that for all time levels n the LT method convergences to the machine zero after i^* iterations, one can immediately conclude that $\left(\widehat{\Lambda}_f^n\right)_{i_*-1} = \left(\widehat{\Lambda}_f^n\right)_{i_*} = \widehat{\Lambda}_f^n$, thus leading to $\widetilde{\Lambda}_f = \left(\widehat{\Lambda}_f^{N_{k+1}}\right)_{i_*-1} = \left(\widehat{\Lambda}_f^{N_{k+1}}\right)_{i_*} = \widehat{\Lambda}_f^{N_{k+1}}$ for $n = N_{k+1}$. Since the term $\widetilde{\Lambda}_f$ in Eq. (16) converges to $\widehat{\Lambda}_f^{N_{k+1}}$ for $1 \leqslant k \leqslant K-1$, the result is that the set of the local adjoint equations defined on each time subinterval converges to the original system of adjoint Eq. (7), provided that the adjoint operators in both systems are the same. Note that if initial values of the design variables for the GT algorithm are set equal to the converged optimal values obtained with the LT method, then the adjoint operators in Eq. (7) are identical to those in Eq. (16). Therefore, the local and global adjoint equations at the extremum point are identical to each other, and one can immediately conclude that the solution of the local-in-time adjoint Eq. (16) is equal to the solution of the global adjoint Eq. (7), thus leading to the equivalence of the corresponding sensitivity derivatives. Taking into account the fact that at the extremum obtained with the LT method, $d\widehat{L}/d\widehat{\mathbf{D}}$ vanishes, the true sensitivity derivative, $dL/d\mathbf{D}$, evaluated at the same point in the design space by using the GT algorithm is equal to $d\widehat{L}/d\widehat{\mathbf{D}}$ and therefore vanishes as well. It implies that the solution obtained with the LT method is optimal with respect to the original optimization problem Eq. (3). Note that in principle, the GT and LT algorithms may converge to different local extrema of the optimization problem Eq. (3). What is important, however, that the solutions computed with both the GT and LT algorithms are local minima of the original optimization problem. It should also be noted that for all test problems presented in the next section, the LT method converges to the same solution obtained with the GT counterpart.

Another attractive feature of the new LT algorithm is that it has the same complexity per optimization cycle as the GT method. Indeed, for the LT algorithm, the flow equations and the corresponding adjoint equations on $(T_{k-1}, T_k]$, $1 \leqslant k \leqslant K$ at each optimization iteration are solved only once. Since there is no overlap between time subintervals, the total number of time steps, over which the LT equations are integrated, is equal to that used for integration of the original adjoint Eq. (7) in the GT algorithm.

The LT algorithm can be directly used for solving both time-dependent optimal control problems whose control variables depend on time and design optimization problems whose design variables are independent of time. This is one of the main advantages of the LT method over the receding horizon technique and its variants (e.g., see [10–12,14]) which are applicable only to optimal control problems, but cannot be directly used for design optimization. Another principle difference between these techniques is that the solution computed with the LT method is a local minimum of the optimization problem (3), while the corresponding solution obtained with any receding horizon technique is only suboptimal with respect to the original minimization problem.

5. Numerical results

We consider design optimization problems governed by the 2-D unsteady Euler equations for supersonic flows in a channel with a bump to evaluate the performance of the new local-in-time method. For all test problems considered, the final time, T_{final}, is set to be 1, and the freestream Mach number is given by

$$M(t) = 2 + 0.1\cos(17\pi t/9). \tag{21}$$

Since the freestream Mach number oscillates in time, the entire flowfield is unsteady. The aerodynamic coefficient in Eq. (4) is chosen to be the time-dependent pressure coefficient at the lower boundary of the computational domain. The bump shape is described by the following equation:

$$y = d_1\psi_1(x) + d_2\psi_2(x) + d_3\psi_3(x),$$

where $\psi_l(x)$, $1 \leqslant l \leqslant 3$ are given polynomials satisfying the requirement that the leading and trailing edges of the bump continuously meet the straight lower wall on either side of the bump. Three coefficients d_1, d_2, and d_3 are design variables, i.e., $\mathbf{D} = [d_1, d_2, d_3]^T$.

The governing equations are discretized by using a first-order, node-centered, finite-volume scheme [20] on structured quadrilateral grids. The inviscid fluxes at cell interfaces are computed using the upwind scheme of Roe [21]. At each time step, the nonlinear discrete flow equations are solved by Newton's method. For each test, the residuals of the 2-D Euler equations and the corresponding adjoint equations are driven below 10^{-12}. The governing equations are integrated over 9 time steps with the nondimensional time step equal to 1/9. Along with the LT method, the SLT version of this algorithm with

Author's personal copy

$\widetilde{\mathbf{\Lambda}}_f = 0$ in Eq. (16) is also considered in the present analysis. For all test problems considered, the number of time subintervals used in the LT and SLT algorithms is set equal to 3 and 9, respectively, and the parameter i_s in the LT method is set to 3. For the SLT algorithm, only the local unsteady flow solution on a current time subinterval is held in the operating memory. For the LT method, in addition to the flow solution on the current subinterval, the adjoint variables $\widetilde{\mathbf{\Lambda}}_f^{N_{k+1}}$ for $1 \leqslant k \leqslant K-1$ are also held in the operating memory, while for the GT method, the entire flow solution history for all time levels is stored. The derivatives of $\mathbf{R}$ and f with respect to $\mathbf{Q}$ and $\mathbf{D}$, which are required to form the adjoint equations and the sensitivity derivative, are calculated by using the complex variable technique developed by Lyness and Moler [23].

First, we validate the implementation of the GT method and compare sensitivity derivatives obtained with the GT algorithm and a forward mode differentiation based on the complex variables approach [23]. The key advantage of the complex variables technique is that for sufficiently small values of the complex step size, this method provides the sensitivity derivative with the machine accuracy, which can be used for validation of the adjoint formulation. For the forward mode differentiation, the complex step size is chosen to be 10^{-10}. Note that at each optimization cycle, the forward mode differentiation technique solves the flow problem as many times as the total number of design variables, while the adjoint-based method requires one solve of the Euler and corresponding adjoint equations per optimization cycle, regardless of the number of the design variables. Table 1 shows the sensitivity derivatives computed with the forward mode differentiation and adjoint methods. As expected, the discrepancy is of the order of round-off error, thus validating the implementation and accuracy of the GT method.

Next, we evaluate the performance of the GT, LT, and SLT methods for the time-dependent design optimization problem (3) and (4) when the target flow is feasible. The feasibility of the target flow implies that there exists a set of design variables in the design space, that recovers the target flow precisely. Note that the value of the objective functional at the extremum is zero, and the optimal design variables are expected to be equal to their exact target values. This problem is well suited for evaluation of the performance of optimization methods, because the exact solution is known and the objective functional vanishes at the extremum. The target pressure coefficient is obtained by solving the unsteady 2-D Euler equations with the design variables chosen to be $d_1 = 0.05$, $d_2 = 0.03$, and $d_3 = 0.01$. The initial value of each design variable is set to be zero, thus initially, there is no bump on the lower wall. The optimization is stopped when the absolute value of the objective functional becomes smaller than 10^{-5}.

Convergence histories of the objective functional obtained with all three algorithms are presented in Fig. 2. Overall, the GT, LT, and SLT methods demonstrate very similar convergence rates. For each method, the value of the objective functional rapidly decreases over the first five iterations, dropping down by almost two orders of magnitude. Then, the convergence rate slows down, and the objective functional gradually decreases until it becomes less than the specified tolerance. Fig. 3 shows convergence histories of all three design variables during the optimization. The most important conclusion that can be drawn from this comparison is that the GT, LT, and SLT methods converge to the same solution. From this standpoint, the solutions obtained with LT and SLT algorithms are optimal with respect to the original optimization problem (3). It should also be noted that all the design variables converge to their target values. From the comparisons presented above it follows that the LT and SLT methods converge to the same optimal solution computed with the GT method, while reducing the storage cost by a factor of 1.5 and 4, respectively. For a larger number of time steps, the storage savings may be considerably higher. As has been pointed out in the foregoing section, the storage cost of the SLT algorithm is independent of the number of time steps and equal to 3 units, where one unit corresponds to memory that is required to store one flow solution vector at each grid point. Note that the SLT method requires the same storage as the steady state adjoint formulation. The storage cost of the GT method is $N + 3$ units and directly proportional to the total number of time intervals, N, while the storage cost of the LT method is $K + N/K + 2$ units, where K is the total number of time subintervals used.

We now evaluate the performance of the LT and SLT methods for minimization of the objective functional defined on a time interval that is smaller than $[0, T_{\text{final}}]$. For this test problem, it is assumed that the objective functional involves only the solution at the terminal time T_{final}, i.e.

$$F_{\text{obj}} = \sum_{j \in \Gamma_c} \left[C_j^N - \left(C_j^{\text{target}} \right)^N \right]^2 \Delta t. \tag{22}$$

The target pressure distribution in Eq. (22) is chosen in the same manner as in the previous test problem. Therefore, the target flow is feasible, and the optimization problem has at least one global minimum. Clearly, this problem is more challenging for the SLT method. Indeed, the SLT method takes into account only the contribution of the last time interval to the sensitivity derivative, while for the GT and LT methods, the adjoint variables at each time level are nonzero; thus, each time subinterval makes a nonzero contribution to the global sensitivity derivative. Fig. 4 shows convergence histories obtained with

Table 1
Sensitivity derivatives computed with the adjoint formulation and the forward mode differentiation based on the complex variable technique.

	$\frac{dL}{dD_1}$	$\frac{dL}{dD_2}$	$\frac{dL}{dD_3}$
Adjoint formulation	−10.5059070229186	−12.2910025055155	−12.8094954127715
Complex variables	−10.5059070229196	−12.2910025055174	−12.8094954127741

Author's personal copy

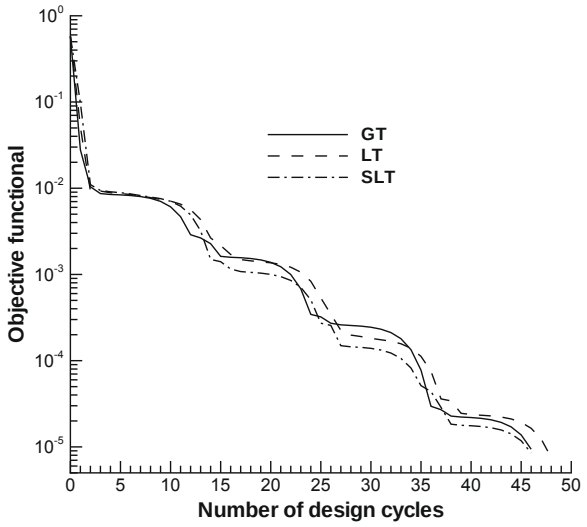


Fig. 2. Convergence of the objective functional obtained with the GT, LT, and SLT methods for the first design optimization problem.

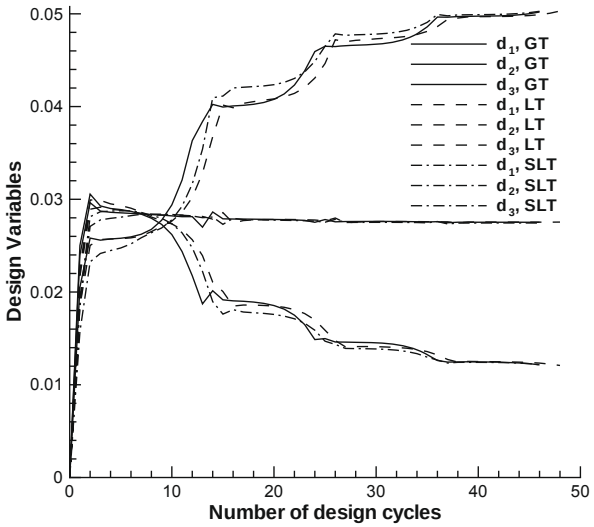


Fig. 3. Convergence of the design variables obtained with the GT, LT, and SLT methods for the first design optimization problem.

the global and both local algorithms for the minimization problem with the objective functional defined by Eq. (22). As follows from Fig. 5, all three methods converge to the global extremum of the minimization problem, demonstrating similar convergence rates. It takes 42 design cycles to reduce the objective functional by four orders of magnitude by using the LT method, while the SLT and GT algorithms require 36 and 37 iterations, respectively.

Despite that for the SLT method, contributions from all time levels except the last one are neglected, its solution and convergence rate are very close to those obtained with GT and LT algorithms. This is not surprising, because as has been shown at the end of Section 3 for this test problem, each component of the sensitivity derivative vector and the flow adjoint variables decay to zero in reverse time. Figs. 6 and 7 demonstrate this property of the sensitivity derivatives and adjoint variables computed with the GT algorithm for the objective functional given by Eq. (22). The result is that the contribution from the last time interval is dominant, which explains why the SLT method provides a good approximation of the total sensitivity derivative. Fig. 7 also shows that the adjoint variables computed with the GT and LT algorithms agree very well over the entire time interval considered, which corroborates our analysis presented in Section 4. Note that for the SLT method, the adjoint equations should be solved only at the final time level, thus reducing the computational cost as compared with the GT and LT algorithms.

For the third test problem, the target bump shape is set to $y = \sin^4(\pi(x-1))$, which is outside of the design space. As a result, the target flow is infeasible, and the value of the objective functional at the optimum is not equal to zero. Fig. 8 shows

Author's personal copy

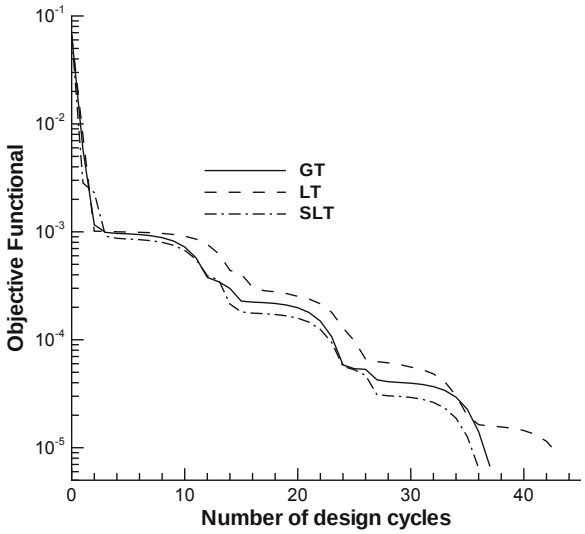


Fig. 4. Convergence histories of the objective functional computed with the GT, LT, and SLT adjoint-based methods for the second test problem.

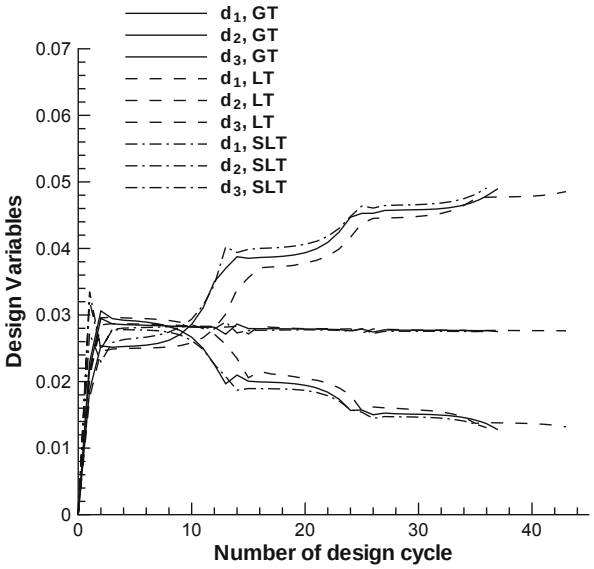


Fig. 5. Convergence of the design variables obtained with the GT, LT, and SLT methods for the second test problem.

convergence histories of the objective functional obtained with the GT, LT, and SLT algorithms. Overall, each optimization method reduces the value of the objective functional more than an order of magnitude.

During the first 15 design cycles, the LT method provides the fastest reduction in the objective functional among all three methods. By 25th design cycle, all the methods provide similar values of the objective functional and show practically the same convergence behavior thereafter. Convergence histories of all three design variables are depicted in Fig. 9. Despite the fact that each design variable changes dramatically during the design, both the SLT and LT methods demonstrate the convergence behavior that is very similar to that of the GT algorithm. As in the previous test cases, the GT, LT, and SLT algorithms converge to the same solution, which again indicates that this solution is optimal with respect to the original minimization problem. The comparison of the computed, target and initial lift coefficients are shown in Fig. 10. The relative difference between the initial lift coefficient and its target value is of the order of $O(1)$. In spite of the fact that the target flow is infeasible, the lift coefficients computed with all three optimization techniques agree reasonably well with the target lift coefficient over the entire time interval considered. Furthermore, the lift coefficients obtained with the GT, LT, and SLT algorithms are almost indistinguishable from each other, which indicates that all three methods converge to the same solution.

Author's personal copy

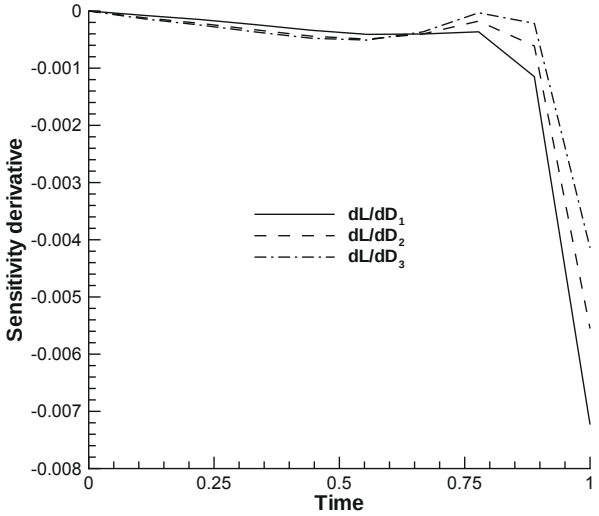


Fig. 6. Components of the sensitivity derivative vector obtained with the GT method.

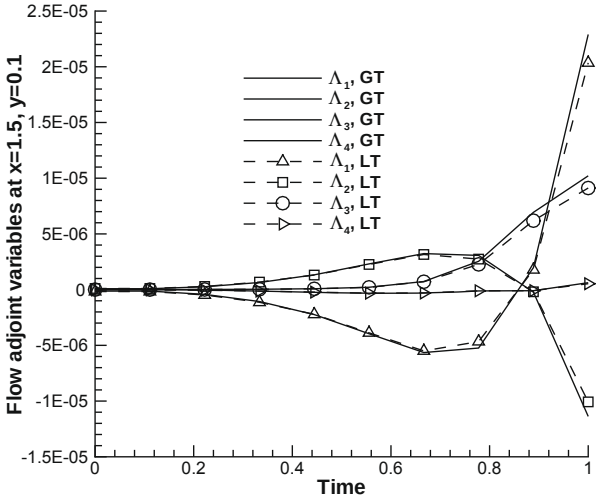


Fig. 7. The flow adjoint variables on the bump surface at $x = 1.5$, obtained with the GT, LT, SLT methods for the second test problem.

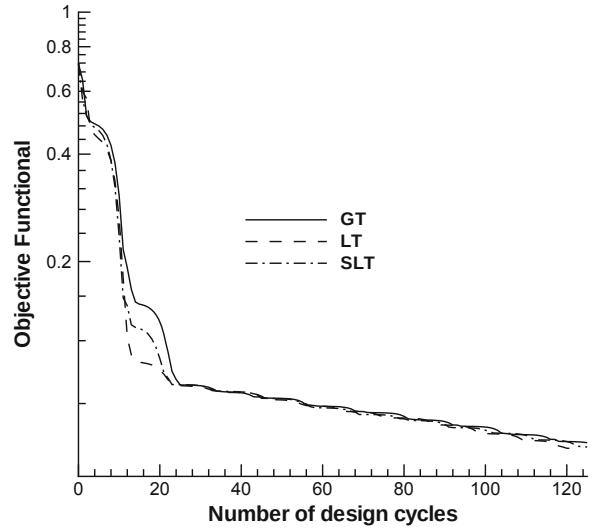


Fig. 8. Convergence histories of the objective functional computed with the GT, LT, and SLT methods for the third test problem (infeasible target flow).

Author's personal copy

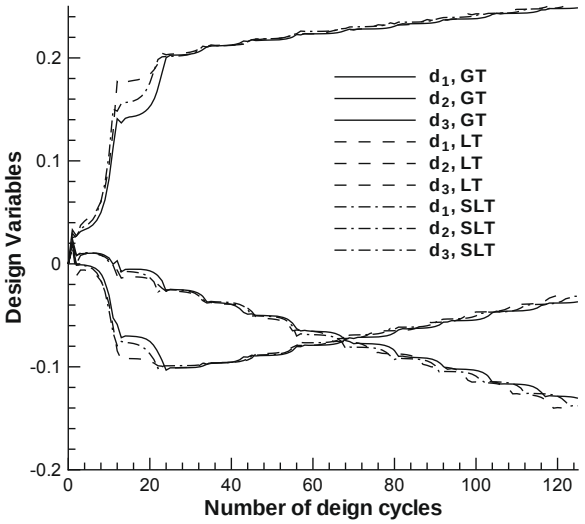


Fig. 9. Convergence of the design variables obtained with the GT, LT, and SLT methods for the third test problem (infeasible target flow).

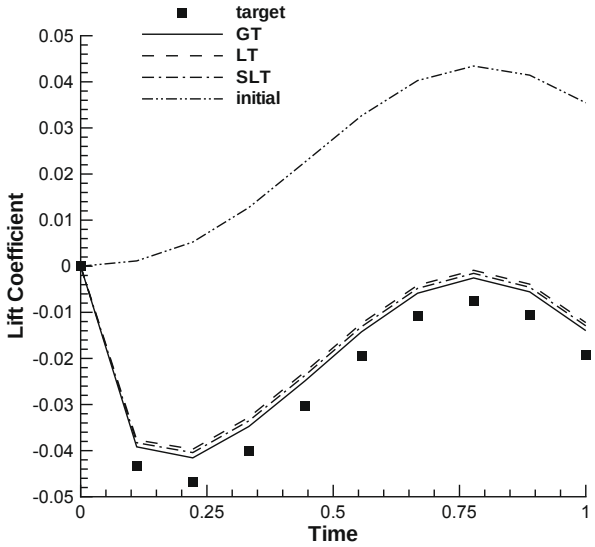


Fig. 10. Comparison of lift coefficients computed using the GT, LT, and SLT methods with their initial and target values for the third test problem (infeasible target flow).

6. Conclusions

The new local-in-time adjoint-based method for design optimization of unsteady flows has been developed. In contrast to the global-in-time (GT) algorithm that stores the flow solution for all time levels, the new algorithm sequentially solves the local adjoint equations on each time subinterval to form the global sensitivity derivative. Two different implementations of the local-in-time method have been considered. The first, simplified (SLT) implementation neglects the coupling between neighboring time subintervals. Since each set of local adjoint equations is integrated backward in time over only a small time subinterval, the storage cost of the SLT method is of the order of $O(N/K)$ flow variables, where N is the total number of time intervals and K is the number of time subintervals. In the limit, each time subinterval can consist of a single time step, thus, the storage cost can be reduced to the level of the steady state adjoint formulation. For the second, more general implementation of the local-in-time (LT) method, the term that couples the local sets of adjoint equations defined on neighboring time subintervals is retained and taken from the previous optimization iteration. The storage cost of the LT method is $O(N/K + K)$ versus $O(N)$ flow variables required for the GT method. For the LT method, the optimal number of time subintervals is $\sqrt{N}$, thus leading to the storage cost that is $\sqrt{N}/2$ times less than that of the conventional counterpart. The most distinctive

Author's personal copy

feature of the LT algorithm is that its solution is a local minimum of the original optimization problem, which is not necessarily the case for the SLT method. Furthermore, for the LT method, the number of operations per optimization cycle is equal to that of the GT algorithm, thus leading to the same CPU cost. For all test problems considered, the GT, LT, and SLT methods provide practically the same convergence rate and converge to the same local minimum of the original time-dependent optimization problem. These properties of the LT method open new avenues for solving a broad spectrum of realistic large-scale design optimization problems arising in various unsteady aerodynamic applications.

Acknowledgments

The work of the first and second authors has been supported by NASA under Contract NNL07AA23C. The first author would also like to acknowledge the partial support from the Army Research Laboratory through Grant W911NF-06-R-006.

References

[1] A. Jameson, N. Pierce, L. Martinelli, Optimum aerodynamic design using the Navier–Stokes equations, Theor. Comput. Fluid Dyn. 10 (1) (1998) 213–237.
[2] E.J. Nielsen, M.A. Park, Using an adjoint approach to eliminate mesh sensitivities in computational design, AIAA J. 44 (5) (2006) 948–953.
[3] E.J. Nielsen, W.K. Anderson, Recent improvements in aerodynamic optimization on unstructured meshes, AIAA J. 40 (6) (2002) 1155–1163.
[4] R. Soto, C, Yang, An adjoint-based methodology for CFD optimization problems, AIAA Paper 2003-0299, 2003.
[5] S.K. Nadarajah, A. Jameson, Optimal control of unsteady flows using a time accurate method, AIAA Paper 2002-5436, 2002.
[6] K. Mani, D.J. Mavriplis, An unsteady discrete adjoint formulation for two-dimensional flow problems with deforming meshes, AIAA Paper 2007-60, 2007.
[7] E.J. Nielsen, B. Diskin, N.K. Yamaleev, Discrete adjoint-based design optimization of unsteady turbulent flows on dynamic unstructured grids, AIAA Paper 2009-3802, 2009.
[8] M. Hinze, A. Walther An optimal memory-reduced procedure for calculating adjoints of the instationary Navier–Stokes equations, in MATH-NM-06-2002, 2002.
[9] M. Hinze, J. Strenberg, A-Revolve: an adaptive memory-reduced procedure for calculating adjoints; with an application to computing adjoints of the instationary Navier–Stokes system, Optim Method Softw 20 (6) (2005) 645–663.
[10] C. Min, H. Choi, Suboptimal feedback control of vortex shedding at low Reynolds numbers, J. Fluid Mech. 401 (1999) 123–156.
[11] T. Bewley, R. Temam, M. Ziane, A general framework for robust control in fluid mechanics, Physica D 138 (2000) 360–392.
[12] L.S. Hou, Y. Yan, Dynamics and approximations of a velocity tracking problem for the Navier–Stokes flows with piecewise distributed controls, SIAM J. Contr. Opt. 35 (6) (1997) 1847–1885.
[13] K.Y. Tang, W.R. Graham, J. Peraire, Optimal control of vortex shedding using low-order models: open loop model development. II: Model based control, Int. J. Numer. Methods Eng. 44 (1999) 945–990.
[14] M. Hinze, K. Kunisch, Three control methods for time-dependent fluid flows, Flow. Turbul. Combust. 65 (2000) 273–298.
[15] X. Cai, F. Ladeinde, A comparison of two POD methods for airfoil design optimization, AIAA Paper 2005-4912, 2005.
[16] A. Hay, J. Borggaard, D. Pelletier, Reduced-order models for parameter dependent geometries based on shape and sensitivity analysis of the POD, AIAA Paper 2008-5962, 2008.
[17] P.A. LeGresley, J.J. Alonso, Investigation of non-linear projection for POD based reduced order models for aerodynamics, AIAA Paper 2001-0926, 2001.
[18] S.K. Nadarajah, M. McMullen, A. Jameson Non-linear frequency domain based optimum shape design for unsteady three-dimensional flow, AIAA Paper 2006-1052, 2006.
[19] S. Choi, M. Potsdam, K Lee, G. Iaccario, J.J. Alonso, Helicopter rotor design using a time-spectral and adjoint-based method, AIAA Paper 2008-5810, 2008.
[20] W.K. Anderson, D.L. Bonhaus, An implicit upwind algorithm for computing turbulent flows on unstructured grids, Comput. Fluid 23 (1994) 1–21.
[21] P.L. Roe, Approximate Riemann solvers, parameter vectors, and differential schemes, J. Comput. Phys. 43 (2) (1981) 357–372.
[22] N.K. Yamaleev, B. Diskin, E.J. Nielsen, Adjoint-based methodology for time-dependent optimization, AIAA Paper 2008-5857, 2008.
[23] J.N. Lyness, C.B. Moler, Numerical differentiation of analytic functions, SIAM J. Numer. Anal. 4 (1967) 202–210. 124–134.

Applied Numerical Mathematics 60 (2010) 224–226

Contents lists available at ScienceDirect

Applied Numerical Mathematics

www.elsevier.com/locate/apnum

Notes on accuracy of finite-volume discretization schemes on irregular grids

Boris Diskin [a,b,*], James L. Thomas [c]

[a] National Institute of Aerospace (NIA), 100 Exploration Way., Hampton, VA 23681, United States
[b] Mechanical & Aerospace Engineering Department, University of Virginia, Charlottesville, VA 22902, United States
[c] Computational Aerosciences Branch, NASA Langley Research Center, Mail Stop 128, Hampton, VA 23681-2199, United States

ARTICLE INFO

Article history:
Received 7 May 2009
Received in revised form 21 November 2009
Accepted 4 December 2009
Available online 11 December 2009

Keywords:
Irregular grids
Accuracy analysis
Truncation error
Discretization error

ABSTRACT

These notes rebut some overreaching conclusions of Svärd et al., 2008 [19] concerning relations between truncation and discretization errors on irregular grids. Convergence of truncation errors severely degrades on general irregular grids. Such degradation does not necessarily imply a less than design-order convergence of discretization errors.

© 2009 IMACS. Published by Elsevier B.V. All rights reserved.

These notes are a response to the recently published article [19]. The article applies a truncation-error analysis to evaluate accuracy of finite-volume discretization (FVD) schemes on general unstructured grids. The analysis is accompanied by computations performed on regular and irregular grids. We consider some of the conclusions overreaching in application to irregular-grid computations.

On regular grids, convergence of *truncation errors* is an accurate indicator of convergence of discretization errors, provided discrete boundary conditions are adequate. However, the truncation-error convergence is often misleading for FVD schemes defined on irregular (e.g., unstructured) grids. As shown in [19] and twenty years earlier in [18], the second-order convergence of truncation errors for some commonly used FVD schemes can be achieved only on grids with a certain degree of geometric regularity. Other studies, e.g., [2–6,9–15,17,20,21], showed that truncation-error convergence degradation on irregular grids does not necessarily imply a degradation of discretization-error convergence. In [13], discretization schemes in which convergence of discretization errors surpasses the convergence of truncation errors were called *supra-convergent* with references dated back to the 1960s [21].

Plentiful computational evidence and a solid body of theory found in the literature demonstrate that on irregular grids, the design-order discretization-error convergence can be achieved even when truncation errors exhibit a lower-order convergence or, in some cases, do not converge at all. Note that these results do not contradict the Lax theorem, which states that consistency (convergence of truncation errors) and stability are sufficient (not necessary) for convergence of discretization errors. While a rigorous proof of discretization error convergence for FVD schemes on general irregular grids is not yet available, there are several recent publications addressing supra-convergence on irregular grids. Eriksson and Nordström [9] analyze one-dimensional (1D) elliptic equations on irregular grids with centered and randomly shifted locations of the dual grid points (flux locations) and prove the discretization-error convergence of orders 2 and 1.5, respectively. Barbeiro [2]

* Corresponding author at: National Institute of Aerospace (NIA), 100 Exploration Way., Hampton, VA 23681, United States.
E-mail addresses: bdiskin@nianet.org (B. Diskin), James.L.Thomas@nasa.gov (J.L. Thomas).

0168-9274/$30.00 © 2009 IMACS. Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.apnum.2009.12.001

proves second-order convergence of discretization errors for formally inconsistent (no truncation-error convergence) discretizations of two-dimensional (2D) elliptic equations on nonuniform grids. Papers [6,17] consider "inconsistent" schemes for advection equations in 1D and 2D and prove convergence of discretization errors. Although we do not show it here, a rigorous proof is in hand for the design-order discretization-error convergence of upwind (and upwind-biased) FVD schemes for constant-coefficient advection equations on random 1D grids. Other discretization-error convergence proofs for some formally inconsistent discretization schemes can be found in Refs. [4,13,21].

Article [19] applied a truncation-error analysis to FVD schemes for the Poisson equation. A "thin-layer" approximation was analyzed. It was shown that the truncation error is $O(1)$ (i.e., does not converge) in grid refinement unless the grids are regular. The discretization error of the scheme was inferred to be non-convergent. By coincidence, the particular thin-layer FVD scheme considered in [19] is indeed zeroth-order accurate even on non-orthogonal structured grids [16]. In [19], a general conclusion was drawn that "a compact finite volume approximation of the Laplacian has to rely on symmetries in the grid to be first-order accurate." This conclusion is incorrect. For example, a common finite-volume scheme equivalent to a Galerkin finite-element approximation (linear elements) on triangles satisfies the definition of a compact scheme and is known to have second-order discretization errors (and zeroth-order truncation errors) on irregular (non-symmetric) grids. FVD schemes for elliptic equations exhibiting similar supra-convergence properties on general mixed-element grids can be found in [8,20].

Article [19] also considered an edge-based *central* FVD scheme for an advection equation on mixed-element and perturbed quadrilateral grids. Truncation-error analysis showed a zeroth-order convergence in the L_∞-norm. Supporting computations showed a zeroth-order convergence of discretization errors. It was concluded that FVD schemes for an advection equation are non-convergent on non-smooth irregular grids. The conclusion is incorrect in general because there are counter examples of FVD schemes with truncation errors that do not converge on general irregular grids but with discrete solutions that converge with at least first order in any norm [8]. The numerical scheme considered in [19] is not representative of current practice—the central scheme is known to exhibit erratic convergence of discretization errors in grid refinement because of lack of h-ellipticity, see, for example [7,8,22]. Note that the article [9] also considers a central scheme for a 1D constant-coefficient advection equation on irregular grids and proves that the mean discretization-error convergence order is at least 0.5, which is better than the zeroth-order convergence predicted in [19] and agrees well with the computational results shown in [8] for a central 2D scheme. For multidimensional advection equations and inviscid compressible and incompressible flow equations, the second-order convergence of discretization errors has been previously demonstrated using upwind edge-based schemes on general simplicial (triangular and tetrahedral) grids; the first-order convergence has been observed on general mixed and perturbed quadrilateral (hexagonal) grids [1,8,20]. The reason for not attaining the design second-order convergence of discretization errors has been traced in [8,20] to the first-order accuracy of control-volume boundary flux integration, which is typical for edge-based FVD schemes on irregular non-simplicial grids.

In summary, degradation of truncation-error convergence does not necessarily imply a lower-order convergence of discretization errors. While the individual computations in [19] appear to be correct, several conclusions derived from a truncation-error analysis regarding degradation of discretization error convergence in irregular-grid computations are overreaching. A vast literature on supra-convergence and substantial computational evidence show that the design-order discretization-error convergence can be achieved even when truncation errors exhibit a lower-order convergence or, in some cases, do not converge at all.

References

[1] M. Aftosmis, D. Gaitonde, T.S. Tavares, Behavior of linear reconstruction techniques on unstructured meshes, AIAA Journal 33 (1995) 2038–2049.
[2] S. Barbeiro, Supraconvergent cell-centered scheme for two dimensional elliptic problems, Appl. Numer. Math. 59 (1) (2009) 56–72.
[3] S. Barbeiro, J.A. Ferreira, R.D. Grigorieff, Supraconvergence of a finite difference scheme in $H^s(0, L)$, IMA J. Numer. Anal. 25 (2005) 797–811.
[4] D. Bouche, J.-M. Ghidaglia, F. Pascal, Error estimate and the geometrical corrector for the upwind finite volume method applied to the linear advection equation, SIAM J. Numer. Anal. 43 (2) (2005) 578–603.
[5] F. De Hoog, D. Jackett, On the rate of convergence of finite difference scheme on nonuniform grids, J. Austral. Math. Soc. Ser. B 26 (1985) 247–256.
[6] B. Despres, Lax theorem and finite volume schemes, Math. Comp. 73 (247) (2003) 1203–1234.
[7] B. Diskin, J.L. Thomas, New factorizable discretizations for the Euler equations, SIAM J. Sci. Comput. 25 (2) (2003) 657–681.
[8] B. Diskin, J.L. Thomas, Accuracy analysis for mixed-element finite-volume discretization schemes, NIA Report 2007-08, National Institute of Aerospace, August 2007.
[9] S. Eriksson, J. Nordström, Analysis of the order of accuracy for node-centered finite-volume schemes, Appl. Numer. Math. 59 (10) (2009) 2659–2677.
[10] J.A. Ferreira, R.D. Grigorieff, On the supraconvergence of elliptic finite difference schemes, Appl. Numer. Math. 28 (1998) 275–292.
[11] P.A. Forsyth, P.H. Sammon, Quadratic convergence for cell-centered grids, Appl. Numer. Math. 4 (1988) 377–394.
[12] M.B. Giles, Accuracy of node-based solutions on irregular meshes, in: D.L. Dwoyer, M.Y. Hussaini, R. Voigt (Eds.), 11th International Conference on Numerical Methods in Fluid Dynamics, in: Lecture Notes in Physics, vol. 323, Springer-Verlag, 1989, pp. 369–373.
[13] H.-O. Kreiss, T.A. Manteuffel, B. Wendroff, J.A.B. White, Supra-convergence schemes on irregular grids, Math. Comp. 47 (176) (1986) 537–554.
[14] D.R. Lindquist, M.B. Giles, A comparison of numerical schemes on triangular and quadrilateral meshes, in: D.L. Dwoyer, M.Y. Hussaini, R. Voigt (Eds.), 11th International Conference on Numerical Methods in Fluid Dynamics, in: Lecture Notes in Physics, vol. 323, Springer-Verlag, 1989, pp. 273–277.
[15] T.A. Manteuffel, A.B. White, The numerical solution of the second-order boundary value problem on nonuniform meshes, Math. Comp. 47 (176) (1986) 511–536.
[16] H. Nishikawa, B. Diskin, J.L. Thomas, A critical study of agglomerated multigrid methods for diffusion, AIAA Paper 2009-4138, in: 19th AIAA CFD Conference, San Antonio, TX, AIAA Journal (2009), in press.
[17] F. Pascal, On supra-convergence of the finite volume method for the linear advection, ESIAM: Proc. 18 (2007) 38–47.
[18] P.L. Roe, Error estimates for cell-vertex solutions of the compressible Euler equations, ICASE Report 87-6, NASA CR-1987-178235, January 1987.

[19] M. Svärd, J. Gong, J. Nordström, An accuracy evaluation of unstructured node-centered finite-volume methods, Appl. Numer. Math. 58 (2008) 1142–1158.
[20] J.L. Thomas, B. Diskin, C.L. Rumsey, Towards verification of unstructured-grid solvers, AIAA Journal 46 (12) (2008) 3070–3079.
[21] A.N. Tikhonov, A.A. Samarsky, Homogeneous difference schemes on nonuniform nets, Zh. Vychisl. Mat. Mat. Fiz. 2 (1962) 812–832; English transl. in: U.S.S.R. Comput. Math. Math. Phys. (1962) 927–953.
[22] U. Trottenberg, C.W. Oosterlee, A. Schüler, Multigrid, Academic Press, London, 2000.

AIAA JOURNAL
Vol. 46, No. 12, December 2008

Toward Verification of Unstructured-Grid Solvers

James L. Thomas*
NASA Langley Research Center, Hampton, Virginia 23681
Boris Diskin†
National Institute of Aerospace, Hampton, Virginia 23666
and
Christopher L. Rumsey‡
NASA Langley Research Center, Hampton, Virginia 23681

DOI: 10.2514/1.36655

New methodology for verification of finite volume computational methods using unstructured grids is presented. The discretization-order properties are studied in computational windows, easily constructed within a collection of grids or a single grid. Tests are performed within each window and address a combination of problem-, solution-, and discretization/grid-related features affecting discretization-error convergence. The windows can be adjusted to isolate particular elements of the computational scheme, such as the interior discretization, the boundary discretization, or singularities. Studies can use traditional grid-refinement computations within a fixed window or downscaling, a recently introduced technique in which computations are made within windows contracting toward a focal point of interest. Grids within the windows are constrained to be consistently refined, allowing a meaningful assessment of asymptotic error convergence on unstructured grids. Demonstrations of the method are shown, including a comparative accuracy assessment of commonly used schemes on general mixed grids and the identification of local accuracy deterioration at boundary intersections. Recommendations to enable attainment of design-order discretization errors for large-scale computational simulations are given.

Introduction

THERE is an increasing reliance on computational simulations in aircraft design practices, supplementing traditional analytic and experimental approaches. Verification and validation methodologies [1] are being developed to ensure the correct application of these simulations. Verification methodologies for structured grids are relatively well-developed in comparison with unstructured grids, especially grids containing mixed elements or grids derived through agglomeration techniques. The summary of the latest of three drag prediction workshops [2] illustrates the problems associated with assessing errors in practical complex-geometry/complex-physics applications. Current practices tend to compare relative errors between methods and experimental results, rather than absolute errors. The motivation for this paper was to advance verification methodologies to predict the code performance in such large-scale computational endeavors.

The verification methodologies proposed here stem from a novel *computational* tool, a *downscaling (DS) test*, for evaluating the accuracy of finite volume discretization (FVD) schemes defined on general unstructured meshes [3]. Performed for a known exact or manufactured solution, the test consists of a series of inexpensive computational experiments that provide local estimates for the convergence orders of the discrete solution (discretization) errors by comparing errors obtained on different scales. The test does not impose any restriction on the grid structure. Analysis methods predicting the performance of DS tests were also developed. The downscaling technique is similar in motivation to the shrinking-grid method of Herbert and Luke [4], but is quite different near the boundaries and does not invoke statistically sampled results.

Traditionally, the discretization accuracy of FVD schemes has been verified by convergence of truncation errors (residuals evaluated with the exact solution). On irregular (unstructured) grids, the DS tests demonstrated, and global grid-refinement computations confirmed, that the discretization accuracy is not directly linked to convergence of truncation errors. In fact, many researchers have observed that convergence of truncation errors is a sufficient, but not a necessary, condition [5–8]. As such, from the standpoint of verification, truncation-error convergence provides a conservative estimate of discretization-error convergence.

The main contribution of the current paper is the use of computational windows to improve verification of unstructured-grid computational methods intended for large-scale applications. In large-scale grid-refinement studies, extensive amounts of data are involved and integral norms often do not provide sufficient information to isolate the source of errors. As an alternative, convergence of discretization errors is studied within computational windows, constructed within a collection of grids or a single grid. The concept of consistent refinement is introduced to allow a meaningful assessment of asymptotic error convergence on unstructured grids. A test performed in each window addresses a combination of problem-, solution-, and discretization/grid-related features affecting discretization-error convergence. The windows can be adjusted to isolate particular elements of the computational scheme (such as the interior discretization, the boundary discretization, or singularities) or tailored to pinpoint regions of interest. Testing can use traditional grid-refinement computations within a fixed window or downscaling, using computations within windows contracting toward a focal point of interest. Also, in DS testing, very small mesh sizes can be used to ensure that testing is within the asymptotic convergence range (where the leading-order terms dominate).

The possible methodologies for verifying convergence of discretization errors on unstructured grids are listed in Table 1. The entries in the table are arranged from highest to lowest computational

Presented as Paper 0666 at the Aerospace Sciences Meeting, Reno, NV, 7–11 December 2008; received 14 January 2008; revision received 11 June 2008; accepted for publication 12 August 2008. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0001-1452/08 $10.00 in correspondence with the CCC.

*Senior Research Scientist, Computational Aerosciences Branch, Mail Stop 128; James.L.Thomas@nasa.gov. Fellow AIAA.
†Senior Research Scientist, 100 Exploration Way; bdiskin@nianet.org; Visiting Professor, Mechanical and Aerospace Engineering Department, University of Virginia, Charlottesville, VA.
‡Senior Research Scientist, Computational Aerosciences Branch, Mail Stop 128; c.l.rumsey@nasa.gov. Associate Fellow AIAA.

Table 1 Verification methodologies

Verification method	Complexity	Interpretation
Grid-refinement computations of discretization errors	Expensive	Precise estimate
DS computations of discretization errors	Fixed cost	Optimistic estimate
Truncation errors via grid-refinement or DS computations	Low	Conservative estimate

cost. Unfortunately, the less expensive estimates are more difficult to interpret correctly. For example, large-scale 3-D grid-refinement computations are quite expensive, but it is quite simple to ascertain attainment of design order in grid refinement if an exact solution is available. DS computations keep the computational costs tractable by reducing the physical size of the domain with succeeding computations, but the DS tests can be overly optimistic in predicting global discretization-error convergence because they do not account for possible error accumulation. For unstructured grids, our experience has been that DS-test predictions of discretization accuracy have been the same as grid-refinement predictions. In any case, because DS tests are always optimistic predictors of discretization-error convergence, if a DS test fails to demonstrate the design performance, there is certainly a deficiency in either the formulation or the implementation. When monitoring truncation errors, the solutions need not be determined, because only residuals need to be evaluated with the manufactured solution. Because this is a local evaluation, there is little difference in convergence orders of truncation errors between grid-refinement and DS tests. Truncation-error assessment is thus inexpensive, but it can serve only as an upper bound, often overly conservative, on discretization-error convergence. This hierarchy of verification tools can be used to complement current practices in large-scale simulations.

The paper is organized as follows. Relations between truncation and discretization errors are discussed first, followed by the definition of consistent refinement with an example. Windowing and downscaling are discussed in the next two sections. Examples are shown for elliptic and inviscid equations, including a comparative accuracy assessment of commonly used FVD schemes on general unstructured grids of mixed type and local accuracy deterioration at boundary intersections using tailored DS tests. Recommendations on verification procedures intended for use within large-scale computational applications are given. The final section contains conclusions.

Discretization and Truncation Errors

The FVD schemes are derived from the integral form of a conservation law:

$$\oint_\Gamma (\mathbf{F}\cdot\hat{\mathbf{n}})\mathrm{d}\Gamma = \iint_V (f-S)\mathrm{d}V \tag{1}$$

where f is a forcing function independent of the solution, S is a solution-dependent source function, V is a control volume with boundary Γ, $\hat{\mathbf{n}}$ is the outward unit normal vector, and $\mathbf{F}$ is the flux vector. The main accuracy measure of any FVD scheme is the *discretization error* E_d, defined as the difference between the exact continuous solution Q to the differential conservation law

$$\nabla\mathbf{F} = f - S \tag{2}$$

and the exact discrete solution Q^h of the discretized Eq. (1):

$$E_d = Q - Q^h \tag{3}$$

A scheme is considered as design-order-accurate if its discretization errors converge with the design order in the norm of interest.

A common approach to evaluate the accuracy of discrete schemes is to monitor the convergence of *truncation errors*. Traditionally, truncation error E_t measures the accuracy of the discrete approximation to the differential Eq. (2) [9,10]. For finite differences, it is found by computing the discrete residuals after substituting the exact solution for the discrete solution. For FVD schemes, the traditional truncation error is usually defined from a time-dependent standpoint [11,12]. In the steady-state limit, after substituting the exact solution Q into the normalized discrete Eq. (1), the truncation error is defined as

$$E_t = \frac{1}{|V|}\left[\iint_V (f^h - S^h(Q))\mathrm{d}V - \oint_\Gamma (\mathbf{F^h}(Q)\cdot\hat{\mathbf{n}})\mathrm{d}\Gamma\right] \tag{4}$$

where $\mathbf{F^h}$ is a reconstruction of the flux $\mathbf{F}$ at the boundary Γ; $|V|$ is the measure of the control volume,

$$|V| = \iint_V \mathrm{d}V \tag{5}$$

f^h and S^h are, respectively, approximations of the forcing function f and the source function S on V; and the integrals are computed according to some quadrature formulas.

Assuming that the discretization error is small compared with the exact solution Q ($|E_d| \ll |Q|$), the discretization error can be evaluated as

$$E_d \approx J^{-1}(Q)E_t(Q) \tag{6}$$

where

$$J(Q) = \frac{\partial}{\partial Q}E_t(Q) \tag{7}$$

is the Jacobian of the truncation-error expression (4).

The traditional definition of truncation error is very useful for structured (regular) grids because the truncation errors converge as $\mathcal{O}(h^p)$ on sequences of refined meshes, where h is a characteristic mesh size and p is the design discretization-accuracy order of the method. For unstructured-grid computations, the convergence of traditional truncation errors is often misleading. Previous studies [6,13–15] noted that second-order convergence of truncation errors for some commonly used FVD schemes can be achieved only on grids with a certain degree of geometric regularity. Examples published elsewhere [3,5–8] and in this paper show that the truncation errors of a design-order scheme can exhibit a lower order of convergence or, in some cases, not converge at all. For some formally inconsistent FVD schemes (traditional truncation errors do not converge), it has been rigorously proven that the discretization errors, in fact, converge [8].

Relation (6) provides the correct order of discretization-error convergence given the truncation-error convergence order. The complexity of evaluation of the discretization-accuracy order rests with evaluation of the inverse Jacobian; as mentioned, truncation errors are easy to compute for a representative manufactured solution. The inverse Jacobian accounts for both interior and boundary discretizations. An example of evaluations of the inverse Jacobian for a formulation focusing on the discrete boundary conditions is given elsewhere [16]. An approximate solution of Eq. (6) using an equivalent linear operator approach has been used to improve the understanding of relations between truncation and discretization errors [3]. Although the approach neglects error-accumulation mechanisms, it can distinguish clearly between inviscid and viscous equations and even between different equations/solution components within a given system.

In this paper, tests are performed for representative manufactured solutions. The manufactured solutions used herein are of two types: either simple analytic functions (collections of polynomials or sines) or exact solutions. The corresponding forcing functions are found by substituting these solutions into the continuous governing equations and boundary conditions. The intent of the approach is to facilitate

testing of discretizations and boundary conditions in situ for large-scale computations; this is possible with slight modifications of most boundary conditions (e.g., evaluating no-slip conditions with a specified wall velocity instead of the typical zero velocity condition). Likewise, in the far field, the exterior conditions are taken from the exact solution rather than from the typical assumption of constant exterior conditions. Not all boundary conditions are amenable to such modifications (e.g., inviscid tangency), and for these we use exact (or manufactured) solutions associated with a particular geometry. An alternative is the mapping construction used by Bond et al. [17].

Consistent Refinement

The general FVD approach requires partitioning the domain into a set of nonoverlapping control volumes and numerically implementing Eq. (1) over each control volume. Two types of FVD schemes are considered: node-centered schemes, in which solution values are defined at the mesh nodes, and cell-centered schemes, in which solutions are defined at the centroids of the control volumes. In the 2-D examples considered here, the primal meshes are composed of triangular and quadrilateral cells; in 3-D computations, the cells are tetrahedral, prismatic, or hexahedral. The *median-dual* partition [18,19] used to generate control volumes for the node-centered discretization is illustrated in Fig. 1 for two dimensions. These nonoverlapping control volumes cover the entire computational domain and compose a mesh that is dual to the primal mesh. For cell-centered FVD schemes, the primal cells serve as control volumes (Fig. 1).

The discrete solution is represented as a piecewise linear function defined within either primal or dual cells. The discretizations are applied on a sequence of refined grids satisfying the *consistent-refinement property*. For global grid refinement, this property requires the characteristic distance across primal and dual cells to decrease consistently with an increase of the total number of degrees of freedom, N. The characteristic distance should tend to zero as $N^{-1/d}$, where d is the number of spatial dimensions. The property enables a meaningful assessment of the asymptotic order of error convergence. In particular, on 3-D unstructured meshes satisfying the consistent-refinement property, the discretization errors of second-order FVD schemes are expected to be proportional to $N^{-2/3}$.

An equivalent mesh size based on the degrees of freedom is defined as $h_N = N^{-1/d}$. An equivalent mesh size based on a characteristic distance is defined in terms of norms of the local control-volume function (i.e., $h_V = \|V^{1/d}\|$, where $\|\cdot\|$ is a norm of choice). For consistently refined meshes, h_V is a linear function of h_N for any computational subdomain (or the entire domain). The assessment of consistent refinement is purely geometric and could be done automatically by inspecting the meshes over local subsets of the domain. Such a technique is envisioned to be most useful during the grid-generation phase to identify and repair regions in which the grids are not consistently refined.

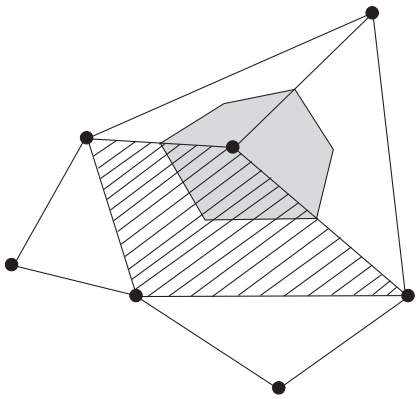


Fig. 1 Illustration of node-centered median-dual control volume (shaded) and cell-centered primal control volume (hashed) in FVD schemes.

To illustrate the concept, we analyze three unstructured tetrahedral grids generated around a sphere; the grids are composed of 25,473 nodes, 82,290 nodes, and 328,463 nodes. In Fig. 2, far-field and near-field views of the coarsest and finest surface grids are shown. In Fig. 3, variations of h_V based on the L_1 and L_∞ norms of $V^{1/3}$ are shown versus the equivalent mesh size $h_N = N^{-1/3}$, each normalized by the value on the coarsest grid. A consistently refined mesh variation is denoted by a dashed line in the figure. Based on the L_1 norm, h_V is linear, but the h_V computed with the L_∞ norm shows that the mesh is not consistently refined. Examination of the grids in Fig. 2 confirms that the mesh near the far-field boundary is not consistently refined. Inviscid incompressible equations for the flow around a sphere have been discretized with a second-order node-centered FVD scheme and solved on these grids. The L_1 norms of the errors in pressure over the field, shown versus h_V in Fig. 4, converge with second order, in spite of the inconsistent refinement. This result is attributed to solution variations being much larger near the surface than near the far-field boundary. Although not shown, we performed computations in a window restricted to a region near the outer boundary and verified that the discretization-error convergence order degrades.

Windowing

To provide a framework for assessing performance of codes in specific large-scale computations, we introduce the concept of windowing. A window is an arbitrarily shaped subdomain within the computational domain serving as a reference frame for testing, and it usually contains a focal point of interest. Figure 5 shows a sketch of possible windows superimposed on an unstructured grid. Solid-line regions are shown with black focal points and dashed-line regions are shown with gray focal points; the latter regions preserve the body geometry (curvature) within the windows. Each test captures an entry from the three groups of features affecting error convergence: 1) problem-related features, 2) solution-related features, and 3) discretization/grid-related features.

The problem-related features are determined by the scope of required computations. Specifically, the features include the interior governing equations, various types of boundary conditions (e.g., inflow, outflow, tangency, no-slip, and symmetry), and the geometrical features characterizing boundaries (e.g., flat boundary, curved boundary, and sharp corners). To address the problem-related features, the windows should be placed in representative locations (interior, boundaries, corners, etc.).

The solution-related features account for variations in the solutions typically encountered, including smooth flows, shocks, stagnation regions, vortices, boundary layers, recirculating flows, etc. Each feature should be represented by a specific choice of the manufactured solution.

The discretization/grid-related features concern variations in meshes and discretization schemes. The features include the interior discretization scheme, discretization of boundary conditions, grid composition [e.g., combinations of advanced-layer (prismatic) regions with interior tetrahedral regions], approximation of geometry (flat panel or higher-order approximation), etc. Interfaces between regions with different types of meshes as well as allowed grid singularities (such as hanging nodes, degenerate cells, etc.) should be considered as separate grid-related features.

Within computational windows, the FVD scheme under study is supplemented with a set of boundary conditions at the interface between the interior and the windowing domain (see the white squares in Fig. 6); overspecification from the known manufactured solution is a typical choice. If the computational window is bounded by a physical boundary, the physical conditions are implemented at the boundary surface; overspecification can still be applied at the remaining interfaces (see the sketches of downscaled windows in Fig. 6). The freedom to choose the manufactured solution, the shape and size of the window, and the type of interface boundary conditions greatly simplifies testing. To verify a code for particular applications,

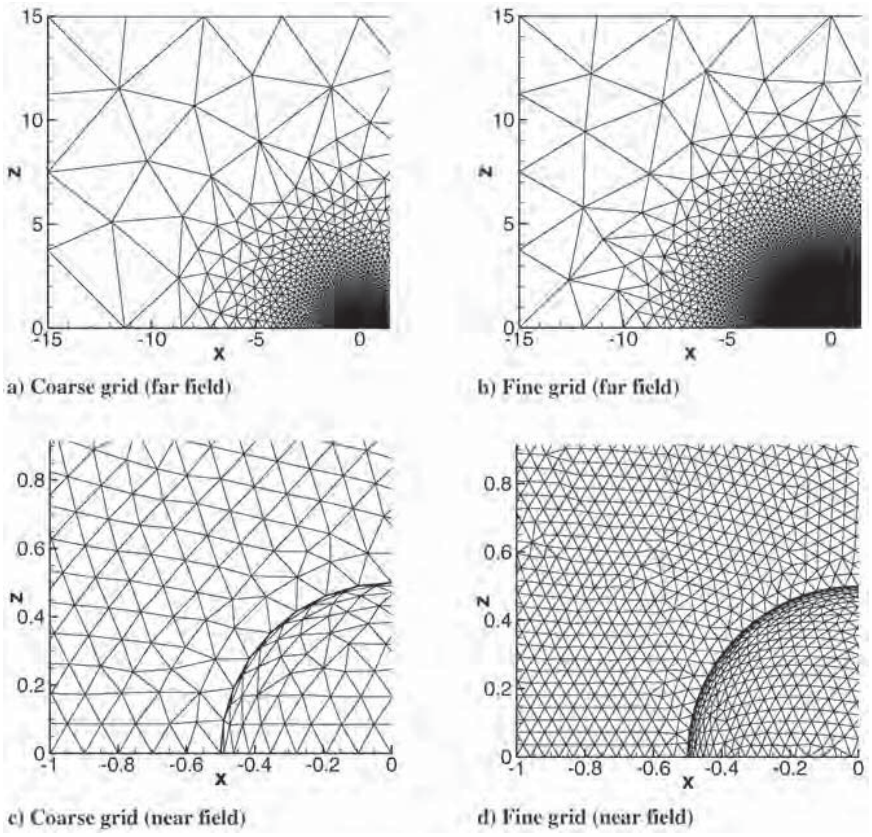


Fig. 2 Partial view of surface grids on symmetry plane and sphere.

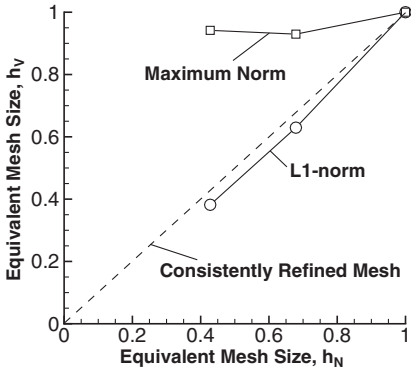


Fig. 3 Consistent-refinement check using normalized equivalent mesh sizes.

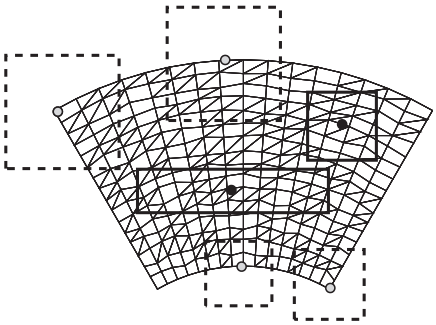


Fig. 5 Sketch of possible windows superimposed on an unstructured grid. Regions denoted by dashed line are windows preserving body geometry (with gray focal points).

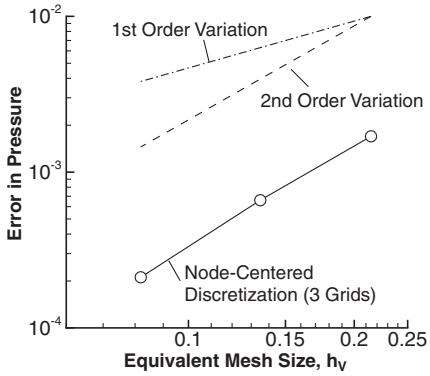
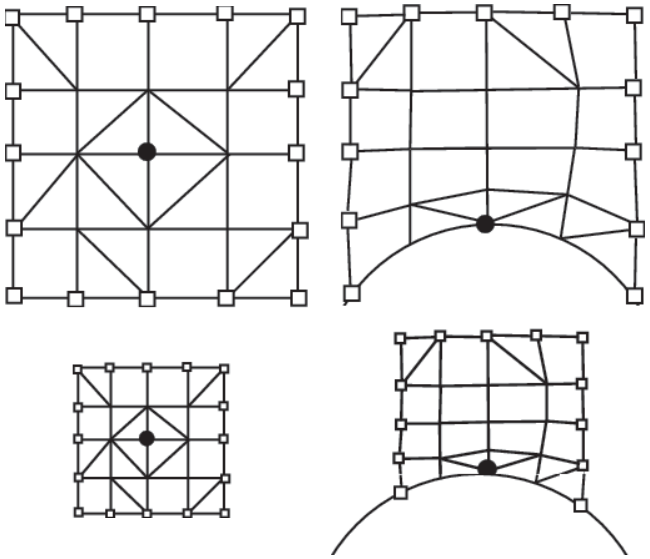


Fig. 4 Variation of L_1 norm of error in pressure over the field with grid refinement.

each representative triplet of features requires a designated test; convergence of discretization errors observed in all representative tests should be understood and accepted as satisfactory.

Downscaling Test

Establishing the discretization-error convergence order in global grid-refinement computations is often not practical because discrete solutions must be computed on grids with prohibitively many degrees of freedom. Constraining the computations to smaller windows makes them more affordable; the DS tests radically reduce the complexity by shrinking domains on grids with smaller mesh sizes, and so the number of degrees of freedom on each grid is kept (approximately) constant. Specifically, the DS test employs numerical computations on a sequence of contracted domains zooming toward a focal point within the original computational domain (Fig. 6). There are at least two possible strategies for grid generation on these contracted domains. The first strategy is termed a



a) Scaledgrid

b) Independent grid generation (accounting for curved physical boundary)

Fig. 6 Illustration of DS computational domains. Black bullets mark the focal points; white squares mark the interface between the interior and the DS-test domain.

scaled grid (Fig. 6a). With this strategy, the first (coarsest) computational domain is defined as a subdomain of the investigated global mesh containing the focal point; other (finer) domains and their mesh patterns are derived by scaling down this first domain [e.g., repeatedly multiplying all the distances from the focal point by a given factor (say, $\frac{1}{2}$ or $\frac{1}{4}$)]. The scaled-grid approach is especially useful for studying interior discretizations and straight boundaries. It is impractical for studies near a general (discretely defined) curvilinear boundary, because the physical boundary shape should be preserved on each grid in the DS sequence. To overcome this limitation, an independent grid (Fig. 6b) can be generated on each domain, assuming a modified consistent-refinement property is satisfied; that is, the characteristic distance across a grid cell is scaled down with the same rate as the diameter of the contracted domains. This second strategy is termed *independent grid generation*.

The DS test evaluates local discretization-error convergence orders by comparing errors obtained in computations on different scales. The tests are performed in all representative computational windows for all representative triplet of features, as described in the Windowing section. The convergence of errors in the L_∞ norm observed in global grid-refinement computations will be bounded by the worst DS-test estimate. Global convergence in integral norms (e.g., L_1 norm) may be better than the worst DS estimate, because these norms are less sensitive to fluctuations occurring locally.

One should interpret the DS-test results carefully because they do not account for possible global discretization-error accumulation. In particular, on structured (regular) grids, convergence of discretization errors observed in DS tests is expected to be a higher order than that observed in grid-refinement computations. In our experience, DS-test estimates of the discretization-error convergence orders on all truly unstructured multidimensional grids (meaning grids with little or no geometric regularity) have been sharp predictors of convergence observed in grid-refinement tests.

In any case, as mentioned earlier, if the convergence of discretization error observed in DS testing is slower than expected, this is an unambiguous indication of deficiencies in either formulation or implementation. Some deficiencies may be found acceptable (for example, when large discretization errors are generated locally and remain local) without affecting integral norms of the errors computed over the entire domain. As an example, for inviscid equations at stagnation, the convergence of discretization errors of velocity components tends to degenerate by one order [3]. This degeneration may or may not be noticed, depending on the flow Reynolds number. Even if observed, the increased discretization error may stay local and not affect convergence of the L_1 norms of the discretization errors.

Example 1: Two-Dimensional Laplace Equation

To illustrate applications of the verification methodology, we first consider the two-dimensional Laplace equation as a model of the diffusion terms in the Navier–Stokes equations,

$$\Delta U = f \tag{8}$$

subject to Dirichlet boundary conditions. The equations are discretized with a second-order node-centered FVD scheme defined on a series of random mixed-element grids composed of triangles and quadrilaterals. The scheme is defined on median-dual control volumes and uses a combination of edge derivatives and Green–Gauss method for evaluating fluxes. Details of the discretization can be found elsewhere [3,20]. The manufactured solution and forcing term are taken as $U = [\sin^2(\pi x) + \sin^2(\pi y)]/2$ and $f = -2\pi^2[1 - \cos^2(\pi x) - \cos^2(\pi y)]$.

For illustration purposes, the computations performed in windows contracted toward the center of the domain are compared with global grid-refinement computations. For global grid refinement, each grid is formed from an underlying structured quadrilateral grid (Fig. 7). In terms of a polar (r, θ) coordinate system, the grid extent is defined as $\theta \in [\pi/3, 2\pi/3]$ in the circumferential direction and $r \in [1, 2.2]$ in the radial direction. The decision to split (or not to split) each structured quadrangle into triangles is determined randomly; approximately half of the quadrilaterals are split. In addition, the interior grid points are perturbed from their original position by random shifts in the range $(-\sqrt{2}/6$ to $\sqrt{2}/6)$ of the local mesh size in the radial direction. The sequences of globally refined grids are generated with $2^{n+3} + 1$ points in both the radial and circumferential directions, where $n = 0, 1, 2, 3, 4$. The sequences of DS grids are generated from a grid with 17 points in both the nominal radial and circumferential directions and downscaled about the center of the domain ($r = 1.6$ and $\theta = \pi/2$) by a factor 2^{-s}, where $s = 0, 2, 4, 6, 8$. The grid topology remains unchanged.

The L_1 norms of truncation and discretization errors are shown in Fig. 8 versus an equivalent mesh-size parameter h_V. Although not shown, error convergence rates in the L_∞ norm are the same as the L_1-norm rates. In grid-refinement computations, the truncation errors remain $\mathcal{O}(1)$ and the discretization errors converge with second order, precisely as predicted by the DS test. The reason for the $\mathcal{O}(1)$ convergence of truncation errors is grid irregularity stemming from the usage of truly unstructured grids. As mentioned previously, the literature frequently associates $\mathcal{O}(1)$ convergence of truncation errors on irregular grids with an indication of an inconsistent scheme that never converges to the exact result; [13,21] this example clearly shows that design-order convergence of truncation errors is not a necessary condition.

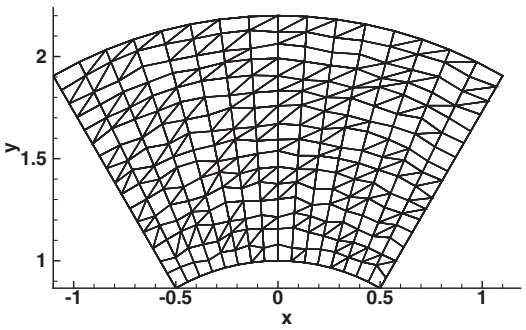
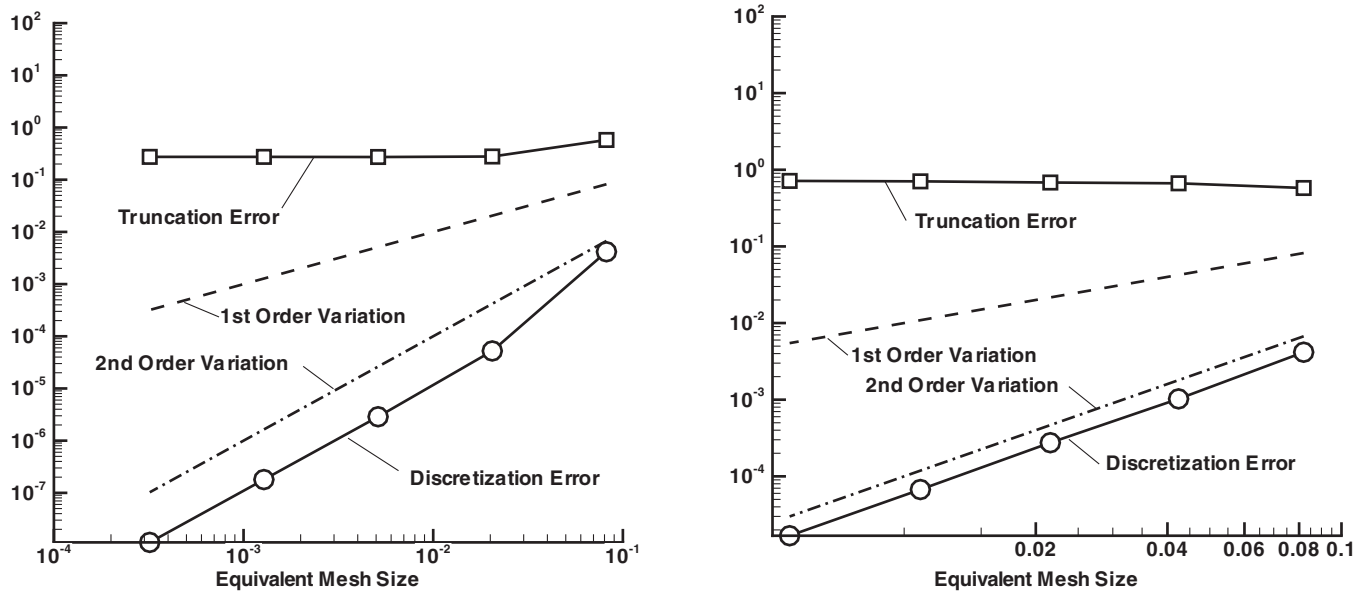


Fig. 7 A typical mixed-element unstructured grid generated with random splitting and random perturbation of the underlying quadrilateral grid.



a) DS test

b) Grid-refinement test

Fig. 8 Convergence of the discretization and truncation errors for the Laplace equation solved on irregular mixed-element unstructured grids.

Example 2: Two-Dimensional Incompressible Euler Equations

In this section, we consider incompressible inviscid equations in the interior and next to the curved tangency boundary. Inviscid fluxes for conservation of mass and momentum are defined as

$$\mathbf{F} = \mathbf{f}\bar{\mathbf{i}} + \mathbf{g}\bar{\mathbf{j}} = \begin{bmatrix} \beta u \\ u^2 + p \\ uv \end{bmatrix}\bar{\mathbf{i}} + \begin{bmatrix} \beta v \\ uv \\ v^2 + p \end{bmatrix}\bar{\mathbf{j}} \tag{9}$$

where the vector of unknowns ($Q = [u, v, p]$) includes the Cartesian velocities and the pressure, and β is an artificial compressibility parameter [20] taken as $\beta = 1$ here.

Two common FVD schemes with design second-order accuracy are investigated: an edge-reconstruction median-dual node-centered scheme and a cell-centered scheme. The node-centered FVD scheme uses the least-squares method for gradient reconstruction and integration over the control-volume boundaries employing split (upwind) fluxes evaluated at the edge medians; details of the discretization can be found elsewhere [3,20]. The cell-centered FVD scheme also employs the least-squares method for gradient reconstruction [18]. Numerical tests are performed for a nonlifting flow around a cylinder of unit radius centered at the origin. The analytical solution for this problem is well known [3].

The first set of tests is performed to study the accuracy of the interior discretization. The computational domain is shifted away from the surface of the cylinder: $1.5 \le r \le 4$ and $2\pi/3 \le \theta \le 4\pi/3$. The two FVD schemes are studied on random triangular and random mixed-element grids. Examples of unstructured grids derived from an underlying structured grid are shown in Fig. 9. Grid randomization is introduced through random splitting (or not splitting) of structured quadrilateral cells. Each cell has equal probabilities to introduce either of the two diagonal choices or, for mixed-element grids, no diagonals.

For each formulation, grid-refinement and DS tests are performed. In global grid-refinement computations, the underlying structured grid is refined by doubling the number of intervals in the radial and angular directions. Randomization is introduced independently on each scale. The inflow boundary conditions are enforced at the boundary corresponding to the external radius; outflow conditions are enforced at all other boundaries. In the DS test, the coarsest 9×9 grid is scaled down around the point $r = 2.75$ and $\theta = \pi$ by multiplying all angular and radial differences from this point by a factor of 0.5. Table 2 summarizes the convergence of discretization and truncation errors observed in these tests. The convergence orders are the same between DS and grid refinement in all norms and for all variables and equations. The results are typical of our experience in comparing DS and grid-refinement tests for unstructured grids.

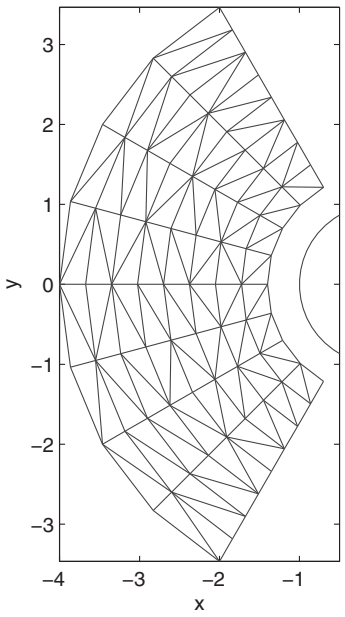
The observed discretization-error convergence rates indicate that the edge-reconstruction node-centered FVD scheme is second-order-accurate on triangular grids, but only first-order-accurate on mixed-element grids; the cell-centered formulation is second-order-accurate on all studied grids. There are many ways to recover second-order accuracy with the node-centered FVD scheme on mixed-element grids. For example, second- and third-order node-centered schemes have been demonstrated with face-reconstruction techniques for flux evaluation [3].

For the edge-reconstruction node-centered scheme, we have also observed first-order convergence of discretization errors with randomly perturbed quadrilateral grids. The results are consistent with a previous publication [22], but contradict another [13]. In the latter reference, $\mathcal{O}(1)$ convergence of discretization errors on randomly perturbed quadrilateral grids with a central scheme was observed. Although not shown, we have implemented a central version of the edge-reconstruction node-centered scheme and tested it for various unstructured grids. We observed first-order convergence of discretization errors on mixed-element and random quadrilateral grids; an in-depth investigation of the discrepancies has been reported elsewhere [3].

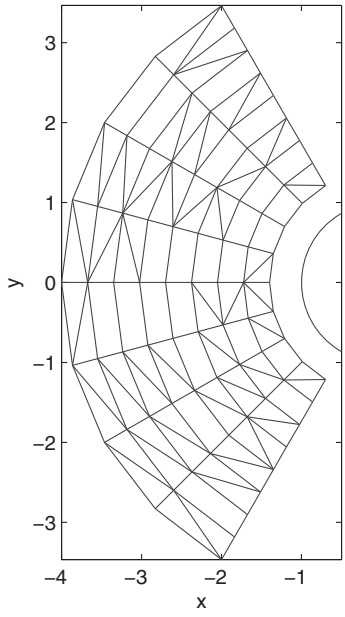
Another series of tests has been performed to study the accuracy of the FVD schemes at the curved tangency boundary; both schemes use isotropic triangular grids approximating the curved tangency boundary by straight segments linking grid nodes located at the physical boundary. The approximation is illustrated in Fig. 10a. The discrete tangency condition is enforced weakly over the straight segments.

A sequence of random triangular grids is generated at the top of the cylinder ($1 \le r \le 2.2$ and $\pi/3 \le \theta \le 2\pi/3$); a grid example is shown in Fig. 10b. Figure 11 illustrates convergence of the L_1 norm of truncation and discretization errors in DS tests performed with the node-centered edge-reconstruction FVD scheme. Figure 11a exhibits convergence observed in the DS test with the focal point in the middle of the tangency boundary; Fig. 11b shows results for the DS test with the focal point next to the inflow/tangency corner. See the sketches in Fig. 11, in which the open squares denote boundaries with overspecification.

Convergence deterioration is clearly observed in the DS test performed with the inflow/tangency boundary conditions, indicating



a) Random triangular



b) Random mixed

Fig. 9 Typical unstructured grids for a computational domain shifted away from the surface of the cylinder.

local loss of second-order accuracy. This local accuracy deterioration is explained and repaired elsewhere [3]. Although not shown, the L_1 norms of the discretization errors in the corresponding grid-refinement test show the second-order convergence, whereas the L_∞ norms of the errors converge with first order. These tests can serve as examples that local accuracy deterioration can be acceptable if the cause and effect on discretization errors are fully understood. Analogous DS tests (not shown) performed for the cell-centered FVD scheme yielded second-order convergence of discretization errors at the interior tangency and at the inflow/tangency corner.

Example 3: Two-Dimensional Compressible Euler Equations

In this section, we solve the compressible Euler equations for the flow over the smooth bump in a channel considered previously by Casper et al. [23]. Using a sheared Cartesian grid mapping, sequences of quadrilateral grids were generated. Mixed-element grids were generated by randomly splitting half of the quadrilateral elements into two triangular elements; the mixed-element grid with 41 and 25 points in the longitudinal and vertical directions, respectively, is shown in Fig. 12. Quadrilateral-element and mixed-element computations are shown for both node-centered and cell-centered formulations. Both formulations use a least-squares method for gradient reconstruction and an approximate flux-difference-splitting scheme.

Tangency boundary conditions were applied on the upper and lower walls, and freestream conditions corresponding to a Mach number of 0.3 were specified at the upstream and downstream locations. In this formulation, the approximate Riemann solver identifies appropriate inflow and outflow fluxes and a tare drag results, attributable to vorticity introduced at the upstream boundary. With an infinitely long channel, the tare drag asymptotes to zero.

Grid-refinement computations are shown in Fig. 13 of the drag minus the tare drag contribution of an infinitely refined mesh. The finest grid contained 641 and 385 points in the longitudinal and vertical directions, respectively. Both quadrilateral-element computations show a third-order variation in the integral measure of net drag. Although not shown, comparison of entropy errors, similar to the technique used by Casper et al. [23], verified that the computations are second-order-accurate. The mixed-element cell-centered computation is second-order-accurate. The mixed-element node-centered computation is only first-order-accurate because of the median-dual approximation of the flux. Windowing computations in the interior of the mesh, not shown, accurately predicted the lower-order behavior of the median-dual approximation for the node-centered mixed-element meshes.

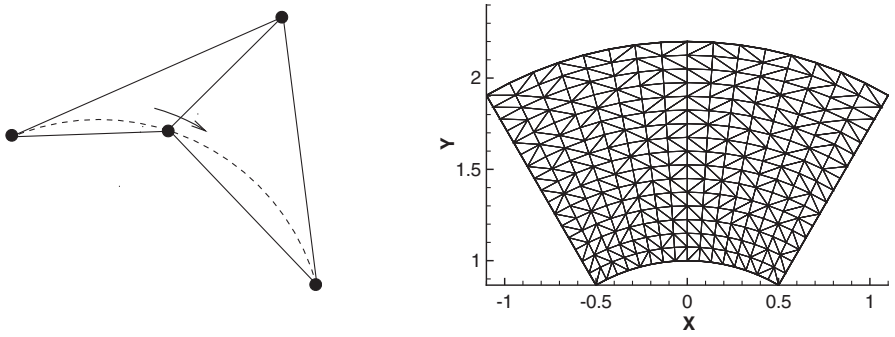
Recommendations on Verification Procedure

In this section, we provide recommendations on choosing relevant tests to verify a code for a large-scale computation; the illustrative examples are motivated by the recent drag prediction workshops [2].

There are two preliminary tests concerned with truncation-error computations (no need to compute discrete solutions), which are useful for confirming consistency of the investigated FVD scheme. The first test is performed for a smooth manufactured solution at fully interior discretizations on *regular-structured*, consistently refined meshes; design-order convergence of truncation errors is expected. The second test is performed for a conservation law equation and a manufactured solution that produces linear fluxes: for example, mass conservation with constant density and linear velocity variations, or momentum conservation with constant density, constant velocity, and linear pressure variations. Second-order (or higher) FVD

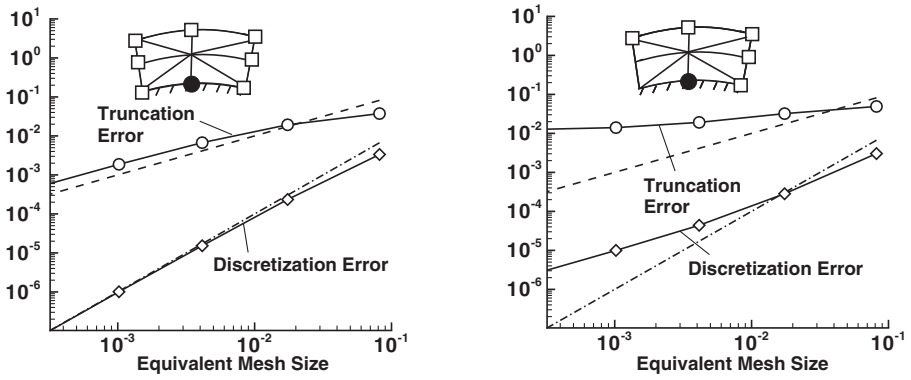
Table 2 Convergence of discretization and truncation errors for various unstructured-grid formulations of the 2-D inviscid incompressible equations on an inflow/outflow computational domain

Formulation	Downscaling computations		Grid-refinement computations	
	Truncation error	Discretization error	Truncation error	Discretization error
Node-centered, random triangular grid	$\mathcal{O}(h)$	$\mathcal{O}(h^2)$	$\mathcal{O}(h)$	$\mathcal{O}(h^2)$
Node-centered, mixed-element grid	$\mathcal{O}(1)$	$\mathcal{O}(h)$	$\mathcal{O}(1)$	$\mathcal{O}(h)$
Cell-centered, random triangular grid	$\mathcal{O}(h)$	$\mathcal{O}(h^2)$	$\mathcal{O}(h)$	$\mathcal{O}(h^2)$
Cell-centered, mixed-element grid	$\mathcal{O}(h)$	$\mathcal{O}(h^2)$	$\mathcal{O}(h)$	$\mathcal{O}(h^2)$



a) Straight-segment approximations to curved tangency boundary (dashed line) **b) Random triangular grid around the top of the cylinder**

Fig. 10 Boundary approximation and grids for DS test of local boundary conditions.



a) DS test: interior tangency boundary condition **b) DS test: inflow/tangency boundary conditions**

Fig. 11 Convergence of the L_1 norm of x-momentum truncation errors and discretization errors in u observed in DS tests performed on random triangular grids surrounding the top tangency boundary of the unit cylinder; dashed and dashed-dotted lines denote first- and second-order error variations; open squares denote boundaries with overspecification.

schemes are expected to exhibit zero truncation errors for equations associated with linear fluxes on *any* mesh.

Assuming the FVD scheme passed these consistency tests, the first step toward forming a library of tests is to formulate a list (as complete as possible) of relevant problem-, solution- and discretization/grid-related features. The following list has been compiled for a mixed-element unstructured-grid solver considered for computations of a viscous flow around an airfoil.

1) *Problem-related features* include Navier–Stokes equations with a given set of parameters, such as Mach and Reynolds numbers; turbulence model; far-field, symmetry, and no-slip boundary conditions; straight or smoothly curved profiles for the far-field and symmetry boundaries; and smooth and discontinuous boundary profiles for the airfoil surface. Each problem-related feature is addressed by choosing an appropriate computational window.

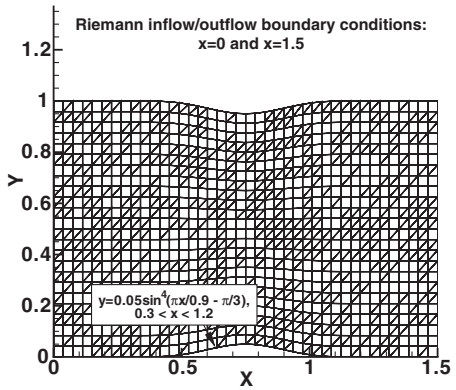


Fig. 12 Mixed-element grid for smooth bump in channel.

2) *Solution-related features* include smooth flow, stagnation flow, vortex, shock, boundary layer, and flow separation. Various solution features are allowed to interact. Each solution-related feature is addressed by choosing an appropriate manufactured solution.

3) *Discretization/grid-related features* include the interior FVD scheme, boundary discretization scheme, advanced-layer prismatic meshes within the boundary layers, and general tetrahedral meshes in the exterior. Interfaces between the regions with different meshing and mesh singularities should be considered as separate grid-related

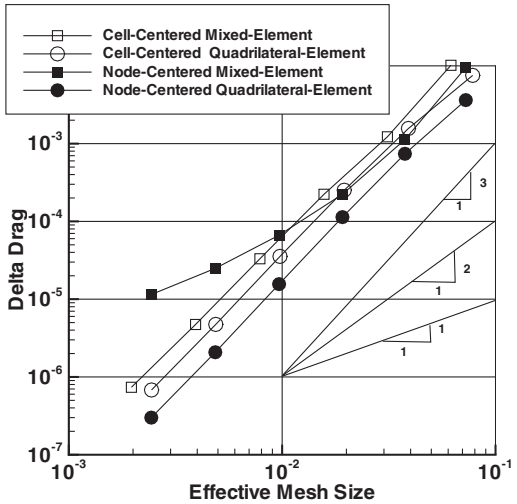


Fig. 13 Comparison of drag variation with effective mesh size for quadrilateral and mixed-element grids for subsonic flow over smooth bump in channel.

features. Each feature is addressed in testing by constructing the grid (grid-refinement generally requires additional grid generation, whereas a DS test may not) and by applying appropriate discrete equations.

A designated test should be designed for each relevant triplet of features, one from each group. Not all triplets are relevant; for example, there is no need to test the combination of a far-field boundary and a boundary-layer solution.

As examples, let us consider the tests recommended for verifying the interior discrete viscous equations (problem-related feature) for smooth solutions away from stagnation (solution-related feature). A computational window is placed away from all physical boundaries and a representative smooth manufactured solution is chosen. In tests performed within this window, second-order convergence of discretization errors is expected. At least four basic combinations of nonsingular meshes should be considered as grid-related features: 1) general prismatic meshes, 2) general tetrahedral meshes, 3) random mixed-element meshes, and 4) meshes with a smooth interface between the prismatic and tetrahedral regions. If certain mesh singularities (e.g., hanging nodes, zero-volume elements, and types of elements other than triangular prisms and tetrahedrons) are allowed, they should be considered in separate tests, usually in combination with the four basic nonsingular meshes.

For verifying the formulation for smooth solutions in the vicinity of a smooth surface, one has to place the window at the surface and perform tests with general prismatic meshes and manufactured solutions representing boundary-layer flow, stagnation flow, and separated flow. For testing smooth solutions around sharply angled parts of the airfoil surface, the same manufactured solutions should be tested on general mixed-element meshes. We have explored only a subset of the recommended practices to date. In particular, the expected asymptotic behavior for discontinuous solutions has yet to be addressed.

Conclusions

New methodology for verification of finite volume computational methods using unstructured grids has been presented. The discretization-order properties are studied within computational windows and address a combination of problem-, solution-, and discretization/grid-related features affecting discretization-error convergence. The windows can be adjusted to isolate particular elements of the computational scheme or tailored to pinpoint regions of interest. Studies can use traditional grid-refinement computations within a fixed window or downscaling, in which computations are made within windows contracting toward a focal point of interest. The only constraint on the grids is that of consistent refinement, enabling a meaningful assessment of asymptotic error convergence on unstructured grids. This concept can be applied to assess families of mapped (block-structured) grids as well. Demonstrations of the method have been shown, including a comparative accuracy assessment of commonly used schemes on general mixed grids and the identification of local accuracy deterioration at boundary intersections. Recommendations to enable attainment of design-order discretization errors for large-scale computational simulations have been given. Perhaps the biggest roadblock to wider usage is that the complete process requires manufactured solutions appropriate to the application and such manufactured solutions are not widely available.

The second possible usage of the accuracy assessment methodology proposed in this paper is in the development of algorithms. Because developments are usually performed in a small-scale environment, demonstrations are simpler than large-scale applications and testing can use both downscaling and grid-refinement approaches relatively easily. Also, appropriate manufactured solutions are easier to construct. Oftentimes, improvements are needed to overcome observed shortcomings of a given scheme and the methodology can be used to pinpoint deficiencies and demonstrate improved capability. A buildup procedure can be used to verify elements of a proposed scheme in a methodical fashion, from interior residual discretizations to boundary residuals. Although we do not emphasize it here, we have found the overall process to be useful in developing efficient solvers, as well as discretizations, for unstructured-grid schemes.

Acknowledgments

This research was supported by the National Institute of Aerospace under NASA Fundamental Aeronautics Program through NASA Research Announcement (NRA) contract NNL07AA23C and cooperative agreement NCC1–02043. The three-dimensional results presented were computed within the FUN3D suite of codes at NASA Langley Research Center. The contributions of E. J. Nielsen and J. A. White of NASA in the implementation of the cell-centered discretization within FUN3D are gratefully acknowledged. The authors also thank the reviewers for their constructive comments and suggestions.

References

[1] Roache, P. J., *Verification and Validation in Computational Science and Engineering*, Hermosa, Albuquerque, NM, 1998.

[2] Morrison, J., and Hemsch, M., "Statistical Analysis of CFD Solutions from the Third AIAA Drag Prediction Workshop," AIAA Paper 2007-0254, Jan. 2007.

[3] Diskin, B., and Thomas, J. L., "Accuracy Analysis for Mixed-Element Finite Volume Discretization Schemes," National Inst. of Aerospace TR 2007-8, Hampton, VA, Aug. 2007.

[4] Herbert, S., and Luke, E. A., "Honey, I Shrunk the Grids! A New Approach to CFD Verification Studies," AIAA Paper 2005-685, Jan. 2005.

[5] Manteuffel, T. A., and White, A. B., Jr., "The Numerical Solution of Second-Order Boundary Value Problems on Nonuniform Meshes," *Mathematics of Computation*, Vol. 47, No. 176, 1986, pp. 511–535. doi:10.2307/2008170

[6] Giles, M. B., "Accuracy of Node-Based Solutions on Irregular Meshes," *11th International Conference on Numerical Methods in Fluid Dynamics*, Lecture Notes in Physics, Vol. 323, Springer, Berlin, 1989, pp. 273–277. doi:10.1007/3-540-51048-6_40

[7] Aftosmis, M. J., and Berger, M. J., "Multilevel Error Estimation and Adaptive h-Refinement for Cartesian Meshes with Embedded Boundaries," AIAA Paper 2002-0863, Jan 2002.

[8] Despres, B., "Lax Theorem and Finite Volume Schemes," *Mathematics of Computation*, Vol. 73, No. 247, 2003, pp. 1203–1234. doi:10.1090/S0025-5718-03-01618-1

[9] Tannehill, J. C., Anderson, D. A., and Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*, 2nd ed., Hemisphere, Washington, DC, 1997.

[10] Hirsch, C., *Numerical Computation of Internal and External Flows*, Vol. 1, Wiley, NY, 1988.

[11] Turkel, E., "Accuracy of Schemes with Nonuniform Meshes for Compressible Fluid Flows," *Applied Numerical Mathematics*, Vol. 2, No. 6, 1986, pp. 529–550. doi:10.1016/0168-9274(86)90006-1

[12] Syrakos, A., and Goulas, A., "Estimate of the Truncation Error of Finite Volume Discretization of the Navier-Stokes Equations on Collocated Grids," *International Journal for Numerical Methods in Fluids*, Vol. 50, No. 1, Jan. 2006, pp. 103–130. doi:10.1002/fld.1038

[13] Svärd, M., and Nordström, J., "An Accuracy Evaluation of Unstructured Node-Centered Finite Volume Methods," *Applied Numerical Mathematics*, Vol. 58, No. 8, 2008, pp. 1142–1158; also NASA CR-2006-214293, Apr. 2006

[14] Roe, P. L., "Error Estimates for Cell-Vertex Solutions of the Compressible Euler Equations," ICASE Rept. 87-6, Hampton, VA, and NASA CR-1987-178235, Jan. 1987.

[15] Lindquist, D. R., and Giles, M. B., "A Comparison of Numerical Schemes on Triangular and Quadrilateral Meshes," *11th International Conference on Numerical Methods in Fluid Dynamics*, Lecture Notes in Physics, Vol. 323, Springer, Berlin, 1989, pp. 369–373. doi:10.1007/3-540-51048-6_57

[16] Diskin, B., and Thomas, J. L., "Analysis of Boundary Conditions for Factorizable Discretizations of the Euler Equations," ICASE Rept. 2002-13, Hampton, VA, and NASA CR-2002-211648, May 2002.

[17] Bond, R. B., Ober, C. C., Knupp, P. M., and Bova, S. W., "Manufactured Solutions for Computational Fluid Dynamics Boundary

Condition Verification," *AIAA Journal*, Vol. 45, No. 9, Sept. 2007, pp. 2224–2236.
doi:10.2514/1.28099

[18] Barth, T. J., "Numerical Aspects of Computing High-Reynolds Number Flow on Unstructured Meshes," AIAA Paper 91-0721, Jan. 1991.

[19] Haselbacher, A. C., "A Grid-Transparent Numerical Method for Compressible Viscous Flow on Mixed Unstructured Meshes," Ph.D. Thesis, Loughborough Univ., Loughborough, England, U.K., 1999.

[20] Anderson, W. K., and Bonhaus, D. L., "An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids," *Computers and Fluids*, Vol. 23, No. 1, Jan. 1994, pp. 1–21.
doi:10.1016/0045-7930(94)90023-X

[21] Coirier, W. J., "An Adaptively Refined, Cartesian, Cell-Based Scheme for the Euler and Navier-Stokes Equations," NASA TM-106754, Oct. 1994.

[22] Aftosmis, M., Gaitonde, D., and Tavares, T. S., "Behavior of Linear Reconstruction Techniques on Unstructured Meshes," *AIAA Journal*, Vol. 33, No. 11, 1995, pp. 2038–2049.
doi:10.2514/3.12945

[23] Casper, J., Shu, C.-W., and Atkins, H., "Comparison of Two Formulations for High-Order Accurate Essentially Nonoscillatory Schemes," *AIAA Journal*, Vol. 32, No. 10, 1994, pp. 1970–1977.
doi:10.2514/3.12240

Z. Wang
Associate Editor

20th AIAA Computational Fluid Dynamics Conference
27 - 30 June 2011, Honolulu, Hawaii

AIAA 2011-3235

Convergence of defect-correction and multigrid iterations for inviscid flows

Boris Diskin* James L. Thomas†

Convergence of multigrid and defect-correction iterations is comprehensively studied within different incompressible and compressible inviscid regimes on medium to high-density grids. Good smoothing properties of the defect-correction relaxation have been shown using both a modified Fourier analysis and a more general idealized-coarse-grid analysis. Single-grid defect correction alone has some slowly converging iterations on grids of medium density. The convergence is especially slow for near-sonic flows and for low Mach numbers. Additionally, the fast asymptotic convergence seen on medium density grids deteriorates on high-density grids. Certain downstream-boundary modes are slowly damped on high-density grids. Multigrid accelerates convergence of the slow defect-correction iterations to the extent determined by the coarse-grid correction. The two-level asymptotic convergence rates are stable and significantly below one in most of the regions but slow convergence is noted for near-sonic and low-Mach compressible flows. The multigrid solver has been applied to the NACA 0012 airfoil and to different flow regimes, such as near-tangency and stagnation. Certain convergence difficulties have been encountered within stagnation regions. Nonetheless, for the airfoil flow, with a sharp trailing-edge, residuals were fast converging for a subcritical flow on a sequence of grids. For supercritical flow, residuals converged slower on some intermediate grids than on the finest grid or the two coarsest grids. At either conditions, convergence of drag below the level of discretization errors occurs in a single cycle.

I. Introduction

Defect correction (DC) is currently a cornerstone approach for solving the Euler and Navier-Stokes equations. Second-order finite-volume discretizations (FVD) require large-stencil linearizations, making direct iterations expensive. Also, linearizations of inviscid discretizations beyond first-order are highly non-positive and difficult to relax. On the other hand, upwind-biased first-order equations are more diagonally dominant and can be relaxed (solved) with conventional approaches. Thus, DC is widely used for second-order solutions,[1] either directly by solving a series of first-order equations with modified residuals or indirectly by using the first-order operator to relax or precondition the second-order equations. The concept is also being applied in p-multigrid methods to solve higher-order discretizations.

Usually, DC is cited as being slow to converge the second-order residuals but fast to converge quantities of engineering interest, such as lift and drag.[1–3] On the other hand, DC has been used to solve large-scale turbulent application problems for many years[4–6] and relatively fast asymptotic convergence of residuals has been observed in many instances. A hierarchical full-approximation scheme (FAS) multigrid method[6,7] with a DC-based relaxation scheme, herein referred to as MG-DC, was previously developed and applied in two dimensions (2D), demonstrating fast convergence of residuals for airfoils at compressible and incompressible conditions.

Analysis of DC convergence for 2D convection has been previously performed in a semi-discrete setting[6,8] in which boundary conditions in one direction are taken into account. A two-level multigrid analysis[6] showed that although the number of cycles to attain convergence was dependent on the mesh density, the dependence was reasonably small and fast asymptotic convergence was eventually attained. A more detailed study of DC alone[8] showed that an asymptotic convergence of about 0.5 per DC iteration is observed in computations. Slow convergent DC iterations may be encountered for nonaligned flows before attaining the asymptotic rate; the number of slow iterations slightly grows on finer grids as $h^{-1/3}$, where h is a characteristic mesh size. This h dependence can be observed for three-dimensional flows as well.

With the current trend of performing complex computations on increasingly larger scales, it is critically important to (re)evaluate performance of traditional algorithms on grids of high density. Analysis of convergence on such grids

*National Institute of Aerospace (NIA), 100 Exploration Way, Hampton, VA 23681, USA, Senior Member AIAA, bdiskin@nianet.org. Also Department of Mechanical and Aerospace Engineering, University of Virginia, Charlottesville, VA 22904, USA. This work is supported by NASA Fundamental Aeronautics Program, Supersonics Project, NRA Contract NNL07AA23C (PI: Prof. N. K. Yamaleev)

†Computational AeroSciences Branch, NASA Langley Research Center, Mail Stop 128, Fellow AIAA, James.L.Thomas@nasa.gov. This work is funded by NASA Fundamental Aeronautics Program, Subsonic Fixed Wing Project

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

has been conducted in this paper. Some surprising results have been obtained regarding the DC asymptotic rate. Specifically, the asymptotic convergence on typical computational grids is significantly different from the asymptotic convergence on high-density grids. The asymptotic rates are essentially invariant for several refined grids of medium density, but the convergence rates slow significantly with progressive grid refinement. The asymptotic slowdown on high-density grids was found first for the Euler system of equations, but was found to occur even for the convection equation alone. The previous results for asymptotic convergence of DC iterations are revisited in the light of these new findings.

The purpose of this paper is to analyze convergence of iterative solvers for inviscid flows, ranging from incompressible to supersonic Mach numbers, to complement the methodology developed previously for diffusion.[9,10] The convergence of the MG-DC algorithm is comprehensively studied within different incompressible and compressible regimes on structured grids of progressively high density. The approach is to first assess the convergence away from any boundaries and discontinuities that may exist and this assessment can be performed using the framework of a small-perturbation (SP) flow. With acceptable and quantified performance within this regime, a solid foundation is established for assessing convergence for the general 2D inviscid flow. The entire flow field around an airfoil, for instance, has at least six distinct regions (regimes): (1) flow away from the boundaries and discontinuities; (2) flow near tangency boundaries away from stagnation; (3) flow within the leading-edge (LE) stagnation; (4) flow within the trailing-edge (TE) stagnation; (5) flow near discontinuities , e.g., shocks; and (6) flow near the far boundary. Each of these flow regimes may introduce difficulties in the multigrid and each should be studied individually, both analytically and computationally.

Several analysis tools are used to characterize performance of the MG-DC scheme. For SP flows, a constant-coefficient approximation is analyzed with the local mode Fourier (LMF) analysis and a semi-discrete (SD) analysis. General quantitative analysis tools[10,11] idealized coarse-grid (ICG) and idealized relaxation (IR), are applied in actual flow computations for assessing multigrid relaxation and coarse-grid correction. The analytical results, confirmed with actual computations, indicate that asymptotic MG-DC convergence rates are stable and well separated from one and are limited on high-density grids by the quality of the coarse-grid correction. The convergence of MG-DC iterations is significantly better than convergence observed in DC iterations alone because multigrid accelerates convergence of slow DC iterations, especially for near-sonic flows and low-Mach compressible flows.

The material in the paper is presented in the following order. For reference, Table 1 includes all acronyms used in the paper. Components of the multigrid and defect correction scheme are presented in Section II. Analysis tools are introduced in Section III. Section IV describes the first-order solver that serves as a driver of the DC iterations. An analysis of DC and MG-DC iterations for SP flows is presented in Section V. Numerical tests and IR/ICG analysis of flows in other regimes are discussed in Section VI. The results are discussed in Section VII. Details of the LMF and SD analysis methods used in this paper are provided in Appendices A and B, respectively. Asymptotic convergence rates of DC iterations on high-density grids are discussed for constant coefficient convection and for SP flows in Appendices C and D, respectively.

II. Components of MG-DC solver

The conservation form of the 2D steady inviscid flow equations is given as

$$\mathbf{R}(\mathbf{Q}) = 0. \tag{1}$$

Here, the conserved variables for compressible flows are $\mathbf{Q} \equiv (\rho u, \rho v, \rho w, \rho, \rho E)^T$, representing the momentum vector, density, and total energy per unit volume, and $\mathbf{R}(\mathbf{Q})$ is a spatial divergence of convective fluxes

$$\mathbf{R}(\mathbf{Q}) \equiv \partial_x \mathbf{F}(\mathbf{Q}) + \partial_y \mathbf{G}(\mathbf{Q}), \tag{2}$$

$$\mathbf{F}(\mathbf{Q}) = \begin{pmatrix} \rho u^2 + p \\ \rho uv \\ \rho u \\ \rho uE + up \end{pmatrix}, \qquad \mathbf{G}(\mathbf{Q}) = \begin{pmatrix} \rho uv \\ \rho v^2 + p \\ \rho v \\ \rho vE + vp \end{pmatrix}.$$

The primitive flow variables are velocity, pressure, and density, $\mathbf{q} = (u, v, p, \rho)^T$. Eq. (1) is discretized with a second-order, cell-centered, upwind-biased FVD scheme that employs an approximate Riemann solver to compute fluxes at the control volume faces. The baseline Riemann solver is the flux-difference-splitting (FDS) scheme[12] but other schemes

Acronym	Description
Alternating Line-Colored (ALC)	A relaxation method
Courant-Friedrichs-Lewy (CFL) number	An iterative parameter characterizing the ratio of (pseudo) time increment to mesh spacing
Correction Scheme (CS)	A MG scheme that uses linear approximations on coarse grids
damped Alternating Line-□acobi (dAL□)	A relaxation method
Defect Correction (DC)	□sed for single-grid iterations and as relaxation in multigrid
Flux Difference Splitting (FDS)	A less-dissipative approximate Riemann solver
Flux Vector Splitting (FVS)	A more-dissipative approximate Riemann solver
Full-Approximation Scheme (FAS)	A MG scheme that uses non-linear approximations on coarse grids
Full-Multigrid (FMG)	A MG scheme that uses coarser-grid solutions to form finer-grid initial approximations
Finite-Volume Discretization (FVD)	The discretization approach used in this paper
Idealized Coarse Grid (ICG)	General quantitative method for analysis of multigrid relaxation
Idealized Relaxation (IR)	General quantitative method for analysis of coarse-grid correction
Leading Edge (LE)	Designate the leading-edge stagnation area
Low-Dissipation Flux-Splitting (LDFS)	A more-dissipative approximate Riemann solver
Local-Mode Fourier (LMF)	A constant-coefficient analysis for interior of the domain, assumes periodicity in all directions
Multigrid (MG)	A hierarchical computational method
MG-DC	Multigrid method studied in this paper that uses defect-correction based relaxation
Semi-discrete (SD)	A constant-coefficient analysis taking boundary conditions into account, assumes periodicity in the directions tangential to the boundary
Small Perturbation (SP)	Computational model that assumes small deviation from a known (e.g., free-stream) solution
Trailing Edge (TE)	Designate the trailing-edge stagnation area

□□□le □. □□ron□ms □se□ □n t□□s p□per.

are also considered, including the low-dissipation flux-splitting (LDFS)[13,14] and flux-vector-splitting (FVS).[15] Either of these latter schemes are generally known to be more dissipative than the FDS scheme. The discrete approximations to derivatives correspond to the Fromm discretization for the structured grids used herein.

The same approach is used for incompressible flows with small variations. The variables are $\mathbf{Q} = (u, v, p)^T$ and the fluxes are defined as in Eq. (2), except the density is constant and the fourth (energy) equation is dropped. The incompressible version of the FDS scheme[4] is used.

In DC, a correction, $\delta\mathbf{Q}^h$, to the approximate solution, $\mathbf{Q}^h$, is computed from the driver equation

$$\mathbf{D}\delta\mathbf{Q}^h = -\mathbf{R}(\mathbf{Q}^h), \tag{3}$$

where $\mathbf{D}$ is the Jacobian of the first-order upwind discretization, and $\mathbf{R}$ is the discretized residual Eq. (1). For DC relaxation within an outer FAS multigrid cycle, a correction scheme (CS) multigrid is applied to determine $\delta\mathbf{Q}^h$. One CS cycle generally reduces the residual of Eq. (3) by an order of magnitude (see Section IV). For individual DC iterations, Eq. (3) is solved to high precision. Where practical, e.g., for the SD analysis or a scalar convection equation, Eq. (3) is solved precisely; otherwise multiple multigrid cycles are used.

After computing $\delta\mathbf{Q}^h$, the solution of the target FVD scheme is updated as

$$\mathbf{Q}^h = \mathbf{Q}^h + \delta\mathbf{Q}^h. \tag{4}$$

For the MG-DC solver, FAS multigrid is used to accelerate convergence. An FAS(ν_1, ν_2) multigrid cycle starts on the target finest grid, performs ν_1 relaxations on the current grid, restricts solutions and residuals to the coarser grid, solves the coarse-grid problem recursively, prolongs the coarse-grid correction, and completes with additional ν_2 relaxations. Each coarse grid is obtained by full coarsening from the finer grid. The same FVD scheme is used on all grids and $W(\nu_1, \nu_2)^1$ cycles are used. For SD computations, the restriction operator is full weighting, and the prolongation operator is the normalized transposition to the restriction. For fully discrete computations, the restriction operator is the conservative residual restriction and prolongation corresponds to linear interpolation. Full multigrid (FMG) requires a high-order prolongation for full efficiency. In the current FMG solver, the FMG prolongation is the same as within the FAS cycle.

III. Analysis tools

In recent years, a number of powerful methods have been developed to analyze convergence of iterative solvers. For problems well described in terms of small perturbations, e.g., SP flows, analysis of a constant coefficient approximation allows one to estimate various convergence characteristics, such as stability, asymptotic and maximum convergence rate, number of slow iterations, etc. For more general problems, windowing and downscaling techniques[16] can be used to analyze accuracy and grid convergence of discrete solutions. Quantitative analysis methods, IR(ν_1, ν_2) and ICG(ν_1, ν_2),[10,11] have proved to be invaluable for assessing components of multigrid solvers for general problems.

III.A. Analysis of Constant Coefficient equations on regular grids

A constant coefficient linearization to the FVD schemes used here on Cartesian grids is given by

$$\mathbf{A}^+\partial_x^- w^h + \mathbf{A}^-\partial_x^+ w^h + \mathbf{B}^+\partial_y^- w^h + \mathbf{B}^-\partial_y^+ w^h = 0, \tag{5}$$

where w^h is a discrete solution vector. For compressible flow, the variables are taken following Mulder[17] as $w^h = (\delta u, \delta v, \delta p/(\rho c), \delta S)^T$, c is the speed of sound, and $S = \log(p/\rho^\gamma)$ is the specific entropy. For incompressible flow, $w^h = (\delta u, \delta v, \delta p)^T$. The operators, ∂_x^- and ∂_y^- are upwind discretizations of derivatives, and ∂_x^+ and ∂_y^+ are downwind discretizations of derivatives.

Different linearizations are associated with each splitting scheme. For the baseline FDS scheme, the linearizations are eigenvalue splittings of the Jacobian matrices associated with non-conservative formulations,

$$\mathbf{A} = \mathbf{A}^+ + \mathbf{A}^-, \qquad \mathbf{B} = \mathbf{B}^+ + \mathbf{B}^-,$$

where

$$\mathbf{A} = \begin{pmatrix} u & 0 & c & 0 \\ 0 & u & 0 & 0 \\ c & 0 & u & 0 \\ 0 & 0 & 0 & u \end{pmatrix}, \qquad \mathbf{B} = \begin{pmatrix} v & 0 & 0 & 0 \\ 0 & v & c & 0 \\ 0 & c & v & 0 \\ 0 & 0 & 0 & v \end{pmatrix},$$

The speed of sound is taken as $c = 1$ and the velocities are defined as $u = M\cos(\alpha), v = M\sin(\alpha)$, where M is Mach number and α is the angle of attack.

For subsonic regimes,

$$\mathbf{A}^+ = \begin{pmatrix} (u+c)/2 & 0 & (u+c)/2 & 0 \\ 0 & u & 0 & 0 \\ (u+c)/2 & 0 & (u+c)/2 & 0 \\ 0 & 0 & 0 & u \end{pmatrix}, \qquad \mathbf{A}^- = \begin{pmatrix} (u-c)/2 & 0 & -(u-c)/2 & 0 \\ 0 & 0 & 0 & 0 \\ -(u-c)/2 & 0 & (u-c)/2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\mathbf{B}^+ = \begin{pmatrix} v & 0 & 0 & 0 \\ 0 & (v+c)/2 & (v+c)/2 & 0 \\ 0 & (v+c)/2 & (v+c)/2 & 0 \\ 0 & 0 & 0 & v \end{pmatrix}, \qquad \mathbf{B}^- = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & (v-c)/2 & -(v-c)/2 & 0 \\ 0 & -(v-c)/2 & (v-c)/2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

For supersonic regimes,

$$\mathbf{A}^+ = \mathbf{A};\ \mathbf{A}^- = 0;\ \mathbf{B}^+ = \mathbf{B};\ \mathbf{B}^- = 0.$$

The discretization is defined as

$$\mathbf{A}^+\partial_x^- w^h + \mathbf{A}^-\partial_x^+ w^h + \mathbf{B}^+\partial_y^- w^h + \mathbf{B}^-\partial_y^+ w^h = 0.$$

In the LMF analysis, the iterations are considered on a periodic domain for discrete Fourier components $w^h = \exp(i(\theta_x i_x + \theta_y i_y))$, where i_x and i_y are integer grid indexes. The Fourier frequencies are normalized: $|\theta_x| \leq \pi, |\theta_y| \leq \pi$. The outcome of the Fourier analysis is an iteration symbol, which is a 4×4 matrix with complex coefficients parametrized by the Fourier frequencies. The specific grid size is reflected through the range of Fourier frequencies realizable on the given periodic grid. For elliptic equations, the maximum spectral radius of the LMF-symbol matrix, taken over all realizable frequencies, is an accurate indicator of the asymptotic convergence rates. For non-elliptic equations, the spectral radius of the LMF symbol is not a sharp estimate for the asymptotic convergence rate on grids of moderate sizes because the LMF analysis accounts only for local error damping, but does not account for boundary effects and error propagation along the characteristics. Note, however, that LMF analysis provides a useful stability test. A larger-than-one LMF spectral radius is an indication of unstable iterations.

For multigrid computations, the relaxation smoothing rate is an important characteristic. The smoothing rate is estimated as the maximum spectral radius of the LMF relaxation symbol, where the maximum is taken over high frequency modes. Typically, high-frequency modes are defined as the modes with $\max(|\theta_x|, |\theta_y|) \geq \frac{\pi}{2}$; all other modes are considered smooth. A more general approach is to define the high-frequency modes as the modes that have relatively large contributions to the residual.[19] An implication of this definition for non-elliptic problems is that the typical set of high-frequencies is reduced: the modes that are smooth in the characteristic directions are excluded, even if their Cartesian frequencies are high. For illustration, for the convection flow at 45° discretized on a uniform Cartesian grid, the mode $\exp(i(\theta_x i_x + \theta_y i_y))$ with $\theta_x \approx \pi$ and $\theta_y \approx -\pi$ is not a high-frequency mode because $\theta_x + \theta_y \approx 0$, and the mode is constant along the characteristic direction. With this modification for non-elliptic problems, the LMF analysis predictions of the smoothing rate are reasonably accurate. A more detailed description of the modified LMF smoothing analysis is provided in Appendix A.

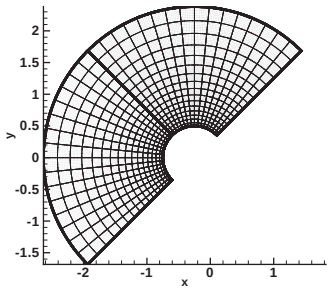
The SD analysis is a good predictor of the asymptotic convergence for non-elliptic problems. The SD analysis assumes solutions in the form $w^h = \exp(i\theta_y i_y)W^h(i_x)$, i.e., the solution is a product of a Fourier component in the y-direction and a discrete function, $W^h(i_x)$, representing solution variations in the x-direction. The SD analysis is accounting for boundary effects and error propagation along the characteristics. For each y-directional Fourier frequency, the asymptotic rate is estimated as the spectral radius of the SD iteration matrix, which has a size proportional to the number of degrees of freedom in the x-direction. Another useful feature of the SD analysis is the capability to identify slow convergent iterations, characterize the error components causing the slow convergence, and explain the mechanism of transition from the slow intermediate convergence to good asymptotic convergence. A detailed description of the SD analysis is provided in Appendix B.

III.□. Gener□l □□□nt□t□t□ve □n□l□s□s

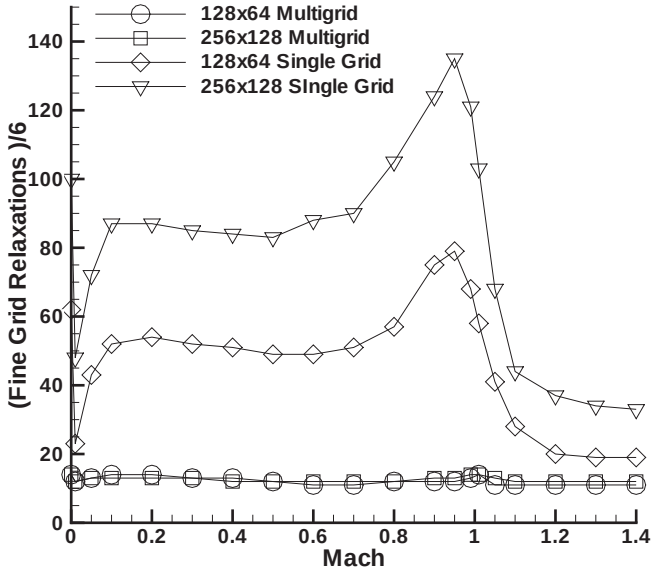
More general, quantitative analysis methods for multigrid solutions are IR and ICG iterations. The iterations are designed to identify slow relaxation or inefficient coarse-grid correction of a multigrid solver. In these iterations, one part of the cycle (coarse-grid correction for IR iterations and relaxation for ICG iterations) is actual and its complimentary part is replaced with an idealized imitation. The IR and ICG methods can be applied to any formulation with a manufactured solution; typically zero solution is used. The initial solution is chosen randomly. In IR iterations, the relaxation in the cycle is replaced with an explicit error averaging procedure. In the IR methods used for this paper, the error at a node is averaged from all the edge-connected neighbors. ICG cycles use actual relaxation scheme and emulate the coarse-grid correction by, first, averaging algebraic errors to the coarse grid and, then, interpolating the averaged error back to the fine grid as a correction. The results of this analysis are not single-number estimates; they are rather convergence patterns of the iterations that may either confirm or refute expectations indicating what part of the actual solver should be improved.

The IR and ICG iterations can be directly applied in the most complicated situations including highly variable (or nonlinear) coefficients, complex geometries, and unstructured grids. The generality of the analysis makes it a valuable tool for analyzing complicated large-scale computational problems, where no other analysis methods are currently available. Properties and specific implementations of IR and ICG methods can be found elsewhere.[10,11]

I□. □□rst-or□er □□ler solver



□□□re □. Com□ne□st□n□t□on □n□t□n□en□□ □r□s.



□□□re □. □□m□er of □er□t□on re□□re□ to re□□□m□□□□ne-□ero res□□□□ls for t□e □rst-or□er solver.

Mulder[2,17] developed efficient 2D multigrid solvers for the first-order upwind discretizations of the inviscid flow equations using both full-coarsening and semi-coarsening approaches. He analyzed many relaxation schemes using a 2-level LMF analysis and showed that the problem of alignment could be addressed uniformly with damped alternating-line-□acobi (dAL□) relaxation within a full-coarsening framework or with point-implicit relaxation within a semi-coarsening framework. In this paper, full-coarsening is used with an alternating-line colored (ALC) relaxation. An under-relaxation factor, $\omega = 0.8$, is needed to effectively smooth high-frequency error.[1,11] The performance of a two-color ALC relaxation is similar to performance of dAL□relaxation.

To illustrate the performance of iterative solvers, computations are performed on a domain around a cylinder. A typical grid is the union of the two cylindrical grids shown in Fig. 1, has local near-unity aspect ratios, and spans 180° of arc sector. Inflow/outflow boundary conditions are applied at all boundaries. The initial solution is a random

perturbation of the uniform free stream conditions.

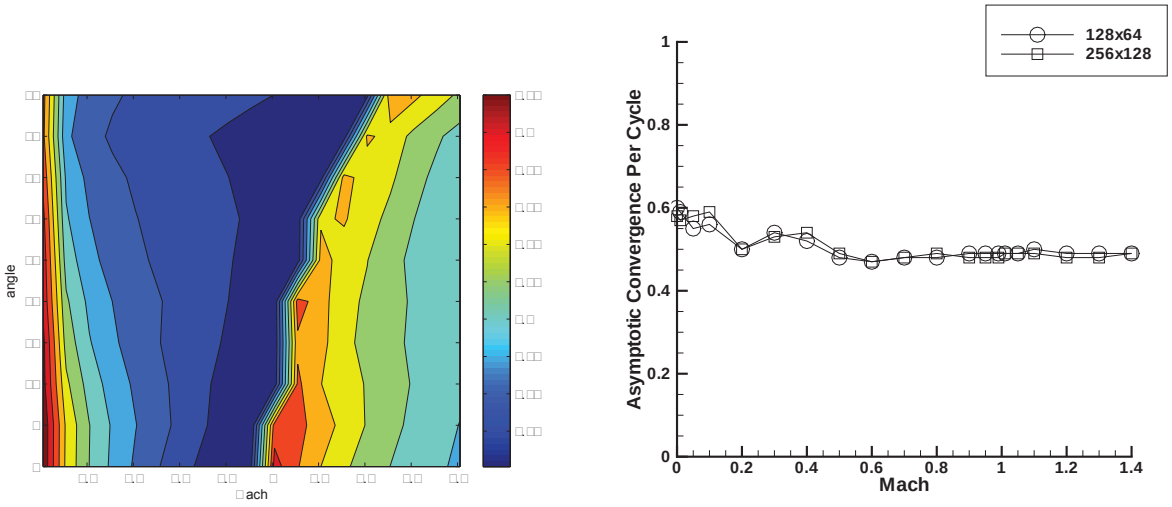
Fig. 2 compares the computational work required to reach the machine-zero residual for single-grid ALC iterations ($\omega = 1.0$) and FAS$(2,1)$ multigrid W-cycles. One ALC iteration is counted as two relaxations and one W-cycle is counted as six relaxations. Results are shown for the FDS scheme on two grids for a range of Mach numbers and for incompressible flow ($M = 0$). The number of single-grid iterations approximately doubles as the grid is refined by a factor of two in each direction, as expected. The required number of iterations is highest at $M \approx 1$. The number of iterations is lowest for the higher Mach numbers and, somewhat unexpectedly, for the least compressible Mach number of $M = 0.01$. The number of fine-grid relaxations observed within MG-DC solver to reach the same residual tolerance is relatively insensitive to variations of Mach number or grid size. Although not shown, the asymptotic MG-DC convergence per cycle is between 0.2 and 0.4 for all Mach numbers on both grids.

C. Multigrid for small-perturbation flows

A previous study showed that, even when the asymptotic convergence rates of DC iterations are fast, a number of slow iterations precedes the asymptotic regime. The slow convergence occurs for smooth characteristic error components[1,19] that are very smooth along the characteristic directions. Such components are removed mainly by accuracy propagation from boundaries along the characteristics. Such removal may take many iterations because an inaccurate driver propagates cross-characteristic oscillations for only shorter distances. Eventually, however, the smooth characteristic errors are removed and asymptotic convergence is attained.

In practical computations, the slow DC iterations may be overlooked on relatively coarse grids because the iterations may arrive to the required solution tolerance before the characteristic components begin to dominate the solution error. In order to observe this slowdown, one should carefully choose the initial solution approximation. On finer grids, this slowdown is a major factor limiting the solution efficiency.

Multigrid accelerates convergence of slow DC iterations. Note that full-coarsening multigrid has its own problems with characteristic components. The asymptotic convergence of the characteristic errors in a two level cycle can be as slow as 0.75 per cycle[19] because cross-characteristic variations propagate shorter distances on coarser grids than on finer grids. The multigrid effects on asymptotic convergence rates are significant only in those flow regimes in which the asymptotic convergence of DC iterations is slower than the coarse-grid correction for characteristic error components. Such situations occur on fine grids.



(a) LMF analysis on 128^2 grid

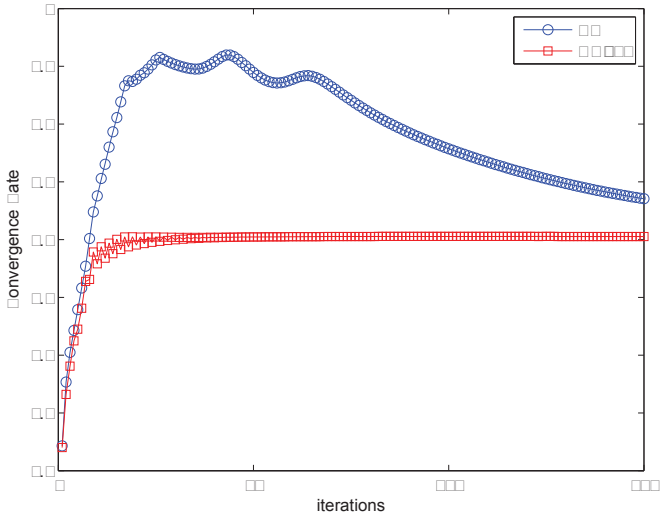
(b) ICG analysis

Figure 3. Smoothing rate of DC iterations.

For a subsequent use in MG-DC cycle, the smoothing rate of DC iterations is estimated with the LMF and ICG analysis. Fig. 3 shows the predicted smoothing rates. For all flow conditions (Mach numbers and angles of attack), the predicted smoothing rates are excellent and grid independent. The LMF predicts the smoothing rate of between 0.5 and 0.7, and the ICG predicts the rate between 0.5 and 0.6. The smoothing rates predicted by ICG are slightly

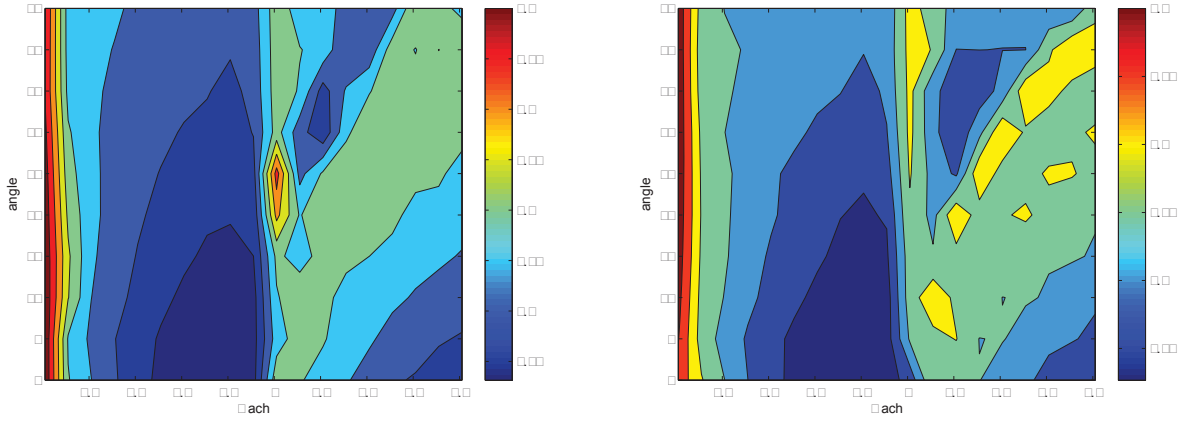
better than the rates predicted by the LMF analysis because ICG predicts the reduction of high-frequency errors in a multigrid cycle, while the LMF analysis predicts the reduction of high-frequency errors in a relaxation. In general, the LMF analysis can be modified to account for the coarse-grid effects on high frequencies.

The need and benefits of multigrid are illustrated in Fig. 4 by the SD analysis for SP flows. The flow conditions are $M = 0.3$, $\alpha = 45°$, the y-directional frequency is smooth $\theta_y = \frac{5\pi}{64}$, and the initial distribution along the x-direction is random. While the asymptotic convergence for both DC and MG-DC iterations is about the same, around 0.6, the slowest convergence rate is significantly slower for DC than for MG-DC iterations.



□□□re □. □D □n□l□s□s □□onver□en□e of DC □n□ MG-DC □ter□t□ons on 256^2 □r□□□ $M = 0.3$ □ $\alpha = 45°$ □□n □ $\theta_y = \frac{5\pi}{64}$.

Fig. 5 shows the asymptotic rates of a two-grid $V(1,0)$ cycle[1] with DC relaxation. The rates are computed with the SD analysis on two coarse grids. The asymptotic rates of MG-DC iterations are stable and well below unity over most of the $M - \alpha$ range. The convergence is slow for near-sonic flows ($M \approx 1$) at intermediate angles of attack and for very low compressible Mach numbers.



(a) Grid 64×64 (b) Grid 32×32

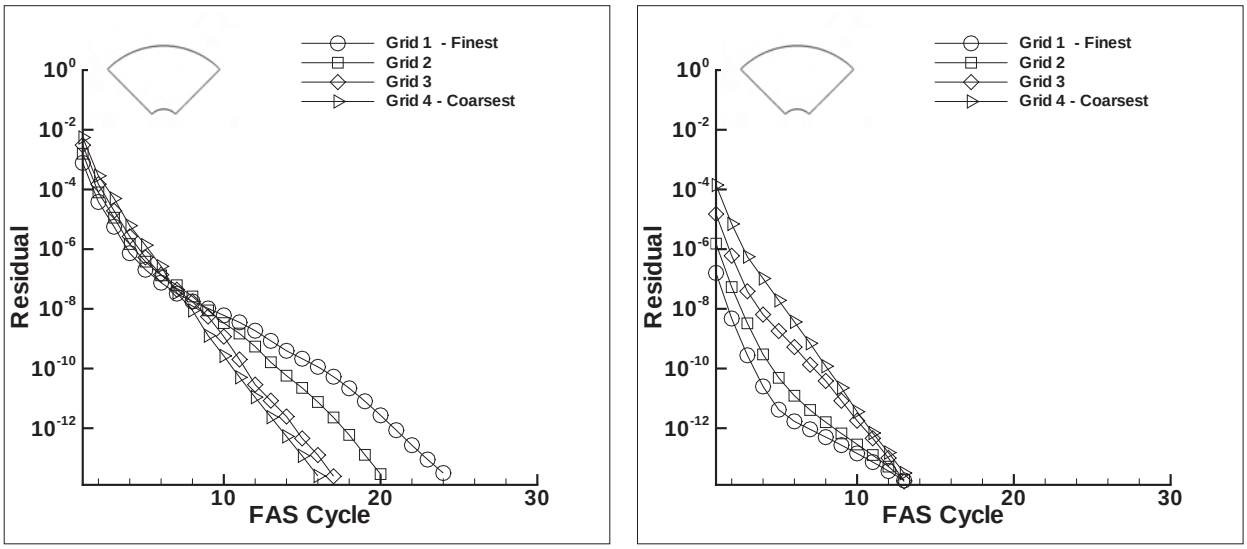
□□□re □. □s□mptot□□ □onver□en□e of t□o-level $V(1,0)$ □□□le □omp□te□ □□t□t□e □D □n□l□s□s. □□e □olor □□rs □re s□□le□ □□□t□e slo□est □s□mptot□□ □onver□en□e o□serve□ on t□e □r□□.

□I. □lo□s □t ot□er re□mes

□I.□. □o□n□□r□-t□n□en□□ □n□st□□n□t□on flo□s

A typical grid for computations of flows characterized by boundary-tangency and LE stagnation was shown in Fig. 1. Stagnation is computed on the most-forward part of the grid and boundary-tangency is computed on the upper-most part of the grid; each domain spans 90 deg of arc sector. A compressible-flow manufactured solution is composed of the velocities from the exact incompressible cylinder flow along with constant enthalpy and entropy. Medium-size grids are considered. The finest grid has 12□cells in both the circumferential and radial directions. Computations are shown for FAS(2,1) W-cycles using a maximum of six levels. Inflow/outflow conditions are applied at all boundaries away from the cylinder surface.

Flows characterized by boundary-tangency do not represent difficulties for the MG-DC solver. A typical convergence history for a series of grids is shown in Fig. 6 for $M = 0.3$, starting from a random perturbation to the exact solution on the left and from FMG interpolations on the right. Starting from random perturbations, the residuals converge rapidly in the first cycles, converge more slowly in intermediate cycles, and then asymptotically converge faster. Starting from FMG interpolations, the number of cycles needed are considerably smaller and machine-level zero residuals are encountered before the faster asymptotic rates are encountered. Although not shown, similar residual convergence per cycle is attained with ICG(1,0) and IR(2,1) multigrid cycles.



(a) Random initial conditions

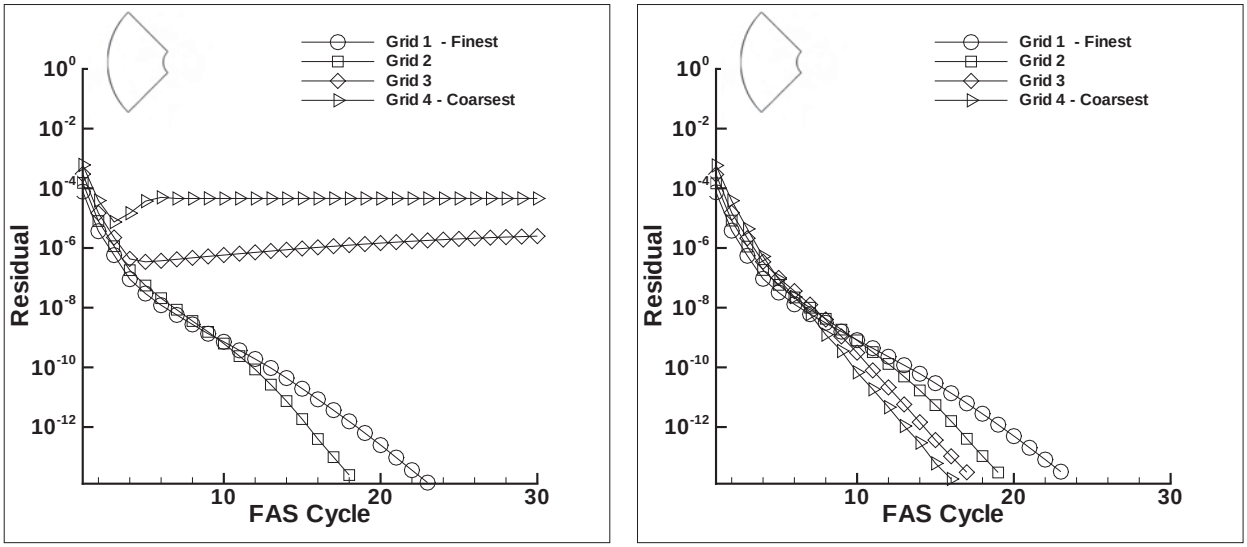
(b) FMG initial conditions

□□□re □. □es□□□l □onver□en□e for □o□n□□r□-t□n□en□□ □omp□t□t□ons □ t□ t□e MG-DC solver□M □ □.□□□D□s□□eme.

Within stagnation flows, the □acobian can differ appreciably from the small perturbation linearization Eq. (5) because the contribution for the velocity gradient (e.g., $O(u_x)$) to the linearization can be comparable with or even greater than the contributions from differences in velocity (e.g., $O(u/h)$). These terms can subtract from the diagonal contributions associated with the momentum equations. For incompressible discretization schemes in which the momentum equations can be marched before solving an elliptic equation for the pressure, these velocity-gradient terms can cause an error amplification when marching into/from stagnation.[11] Here, we find that similar difficulties arise for the MG-DC solver because DC can be unstable. The DC convergence is sensitive to the particular discretization schemes used for LE stagnation. For instance, DC does not converge for the FDS scheme but does for the LDFS and FVS schemes. Fig. 7 shows convergence of the MG-DC solver for stagnation flow using the FDS scheme (left) and the LDFS (right) scheme. An infinite CFL number is used for the LDFS scheme but a CFL of 400 is used for the FDS scheme. □n the two coarser grids, the MG-DC scheme does not converge for the FDS scheme □ a smaller CFL is necessary for the scheme to remain stable. □n the finer grids, the overall residual convergence of either scheme within stagnation is similar to that observed for boundary-tangency computations.

Although not shown, for TE stagnation, both schemes are unstable without addition of a pseudo time step. Analysis of convergence within stagnation leads to a variable-coefficient problem problem that is difficult to analyze using LMF analysis. □ne can devise neighborhoods which provide relevant constant-coefficient approximations to the full

linearization, but certain parts of stagnation, such as the stagnation streamline, are inaccessible to a constant-coefficient analysis.[11] The stagnation flow analysis was actually a motivating factor for the development of more general quantitative analysis methods, such as IR and ICG. For the airfoil computations in the next section, we simply use the LDFS scheme. The airfoil has a sharp trailing edge which does not seem to cause a problem with this scheme.



(a) FDS scheme, CFL = 400

(b) LDFS scheme, CFL = ∞

Figure 8. Stagnation computations residual convergence for the MG-DC solver. Finest grid is 128^2 cells, the coarsest grid is 16^2 cells.

III.C. Multigrid for Airfoil

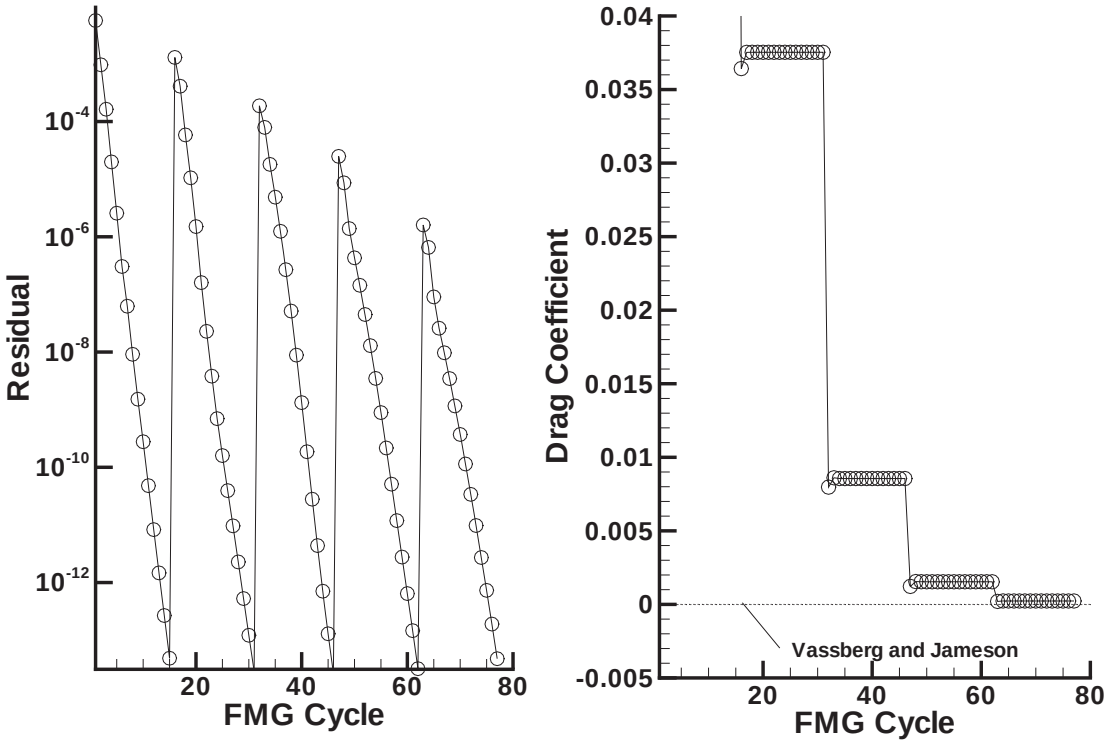
Computations for the NACA 0012 airfoil following the Vassberg and Jameson benchmark study[20] are shown here. The grid, similar to that used for the study, is generated through a sheared adaptation of a conformal grid around a Karman-Trefftz airfoil matching the leading-edge radius and trailing-edge angle of the NACA 0012 airfoil. The grid extends 150 chords outwards from the airfoil, and has nearly unity-aspect-ratio cells. The second-order accuracy was verified in computations with lifting and non-lifting manufactured solutions for the Karman-Trefftz airfoil in incompressible flow and in compressible flow at moderate Mach numbers. A compressible-flow manufactured solution was defined with the velocities from the exact incompressible Karman-Trefftz solution along with constant enthalpy and entropy.

Fig. and Fig. 9 shows residual and drag convergence history of FAS(2,1) cycles for the NACA 0012 airfoil at subcritical lifting conditions ($M = 0.5$ and $\alpha = 1.25$) and supercritical non-lifting conditions ($M = 0.8$ and $\alpha = 0$), respectively. Six grids were used in the computations. FMG cycles were started on the coarsest grid composed of 16^2 cells. The finest grid contained 256 cells in the directions around and outward from the airfoil. For the subcritical computations, convergence rates per cycle are uniformly fast. Residual convergence per cycle is 0.3 on the finest grid. Convergence of drag (and also lift, although not shown) is quite fast, within one FMG cycle. The exact drag is zero, reflected in the benchmark level shown as well as the value on the finest grid. The drag is converging with second order accuracy although finer grids are necessary to confirm this.

For the supercritical computation, convergence rates per cycle are quite disparate between grids. The two grids before the finest grid in the FMG sequence are converging much slower than the finest grid or the two initial coarser grids. No limiter is used in these computations. Drag is again converging within one FMG cycle. The drag is converging with second order accuracy to the benchmark level.

III. Discussions

The MG-DC solver used here is similar to the multigrid scheme developed previously.[6,7] The previous scheme used alternating-line Jacobi and/or colored relaxations that do not provide sufficient damping of high-frequency errors in purely inviscid regions of the flow. Analysis methods were not applied to identify this shortcoming and instead other parts of the algorithm were modified to compensate, namely a pseudo-time step limited by a maximum CFL

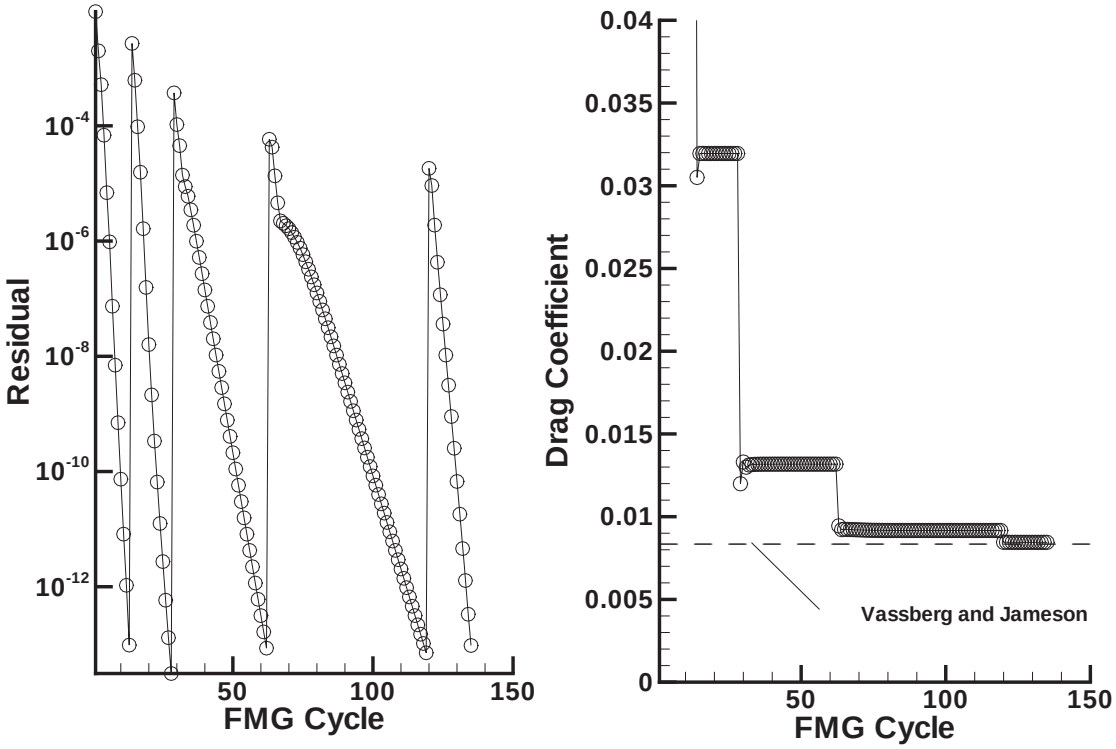


□□□re □. □es□□□□l □n□ □r□□ □onver□en□e □stor□of □□ □□□□□□□□□les for □□C□ □□□□ □□rfo□l □t s□□□r□t□□l lft□n□ □on□t□ons□$M = 0.5$ □n□$\alpha = 1.25$.

of □(100) was added to the implicit relaxation operator, relaxation subiterations were performed, and dissipation via entropy fixes to all fields was added to the FDS discretization. In the present work, we apply under-relaxation based upon optimization of ICG(1,0) cycles, do not add a pseudo-time step except within stagnation, and do not add any entropy fixes.

The convergence of the MG-DC solver has been comprehensively studied within different incompressible and compressible inviscid regimes. The properties of the solver away from any boundaries and discontinuities are analyzed on high-density grids because this region forms the foundation of the methodology. Within this region, the smoothing properties of the scheme have been shown to be bounded away from one using both a modified LMF analysis and a more general ICG analysis. DC alone has some slowly converging iterations on grids of medium density. This behavior has been shown previously for convection but the convergence for the Euler equations is slower than that for pure convection. The convergence is especially slow for near-sonic flows and for very low compressible Mach numbers. Additionally, the asymptotic convergence seen on medium-density grids is significantly different from the asymptotic convergence on high-density grids. Certain downstream-boundary modes are slowly damped on high-density grids. The FAS multigrid scheme accelerates convergence of the slow DC iterations to the extent determined by the coarse-grid correction. The 2-level asymptotic convergence rates are well separated from unity over most of the region but slow convergence is noted for near-sonic and low-Mach compressible flows.

We have applied the MG-DC solver to the NACA 0012 airfoil and to different flow regimes, such as near-tangency and stagnation. The MG-DC solver encounters problems within stagnation regions. The FDS scheme is unstable without a time step addition for leading-edge stagnation and all schemes have a problem for smooth trailing-edge stagnation. Analysis of the linearization within stagnation predicts difficulties associated with the loss of diagonal contributions to the momentum equation linearization within decelerating flow. A pseudo-time step addition can provide convergence, although the amount varies from grid to grid. Nonetheless, for the airfoil flow, with a sharp trailing-edge, residuals were fast converging for a subcritical flow on a sequence of grids. For supercritical flow, residuals converged slower on some intermediate FMG grids than on the finest grid or the two coarsest grids. The



□□□re □. □es□□□l □n□ □r□□ □onver□en□e □stor□of □□ □□□□□□□□les for □□C□ □□□□ □□rfo□□t s□per□r□□□l non-l□ft□n□ □on□□t□ons□$M = 0.8$ □n□ $\alpha = 0$.

cause of the slowdown may be associated with the coarse-grid correction near Mach unity. Also, the lift and drag both showed second-order accuracy in grid refinement for subcritical and supercritical conditions.

A key measure of efficiency for a multigrid method is the number of FMG cycles required to converge algebraic errors below the level of discretization errors. Ideally only a single cycle is needed. For both airfoil solutions, algebraic errors in lift and drag were well below discretization errors after a single FMG cycle. Another key property for an iterative solver is to ensure that the residual can be driven (fast) to the zero level if needed. The MG-DC solver provides fast residual convergence. The efficiency of the scheme is limited by the coarse-grid correction. Previous work has shown that a modified coarse-grid discretization can substantially improve the correction. The effectiveness of the scheme needs to be explored on high-density grids and in the regimes with slower convergence. Local relaxations in slow-convergence regions may accelerate convergence even further.

□. Mo□f□e□ □o□□l Mo□e □o□r□er □n□l□s□s

For given Mach number and angle of attack, the respective symbols of the target, $\mathbf{T}$, and driver, $\mathbf{D}$, operators on a uniform Cartesian grid with mesh spacing h are defined as

$$\begin{aligned}\hat{\mathbf{T}}(\theta_x,\theta_y) \equiv\ & \mathbf{A}^+\tfrac{1}{4h}\left(e^{i\theta_x i_x}+3-5e^{-i\theta_x i_x}+e^{-2i\theta_x i_x}\right)\\ & -\mathbf{A}^-\tfrac{1}{4h}\left(e^{-i\theta_x i_x}+3-5e^{i\theta_x i_x}+e^{2i\theta_x i_x}\right)\\ & +\mathbf{B}^+\tfrac{1}{4h}\left(e^{i\theta_y i_y}+3-5e^{-i\theta_y i_y}+e^{-2i\theta_y i_y}\right)\\ & -\mathbf{B}^-\tfrac{1}{4h}\left(e^{-i\theta_y i_y}+3-5e^{i\theta_y i_y}+e^{2i\theta_y i_y}\right).\end{aligned} \tag{6}$$

$$\begin{aligned}\hat{\mathbf{D}}(\theta_x,\theta_y) \equiv\ & \mathbf{A}^+\tfrac{1}{h}\left(1-e^{-i\theta_x i_x}\right)-\mathbf{A}^-\tfrac{1}{h}\left(1-e^{i\theta_x i_x}\right)\\ & +\mathbf{B}^+\tfrac{1}{h}\left(1-e^{-i\theta_y i_y}\right)-\mathbf{B}^-\tfrac{1}{h}\left(1-e^{i\theta_y i_y}\right).\end{aligned} \tag{7}$$

The symbol, $\hat{\mathbf{DC}}$, of the DC iteration is a 4×4 matrix

$$\hat{\mathbf{DC}}(\theta_x, \theta_y) \equiv \hat{\mathbf{I}} - \hat{\mathbf{D}}^{-1}\hat{\mathbf{T}}, \tag{8}$$

where $\hat{\mathbf{I}}$ is the 4×4 identity matrix.

The smoothing rate, μ, is estimated as the maximum spectral radius

$$\mu = \max \rho \left(\hat{\mathbf{DC}} d \right), \tag{9}$$

where d is a high-frequency indicator. For a flow with $\alpha \leq 45°$,

$$d = \begin{cases} 1, & \text{if } \max(|\theta_x|, |\theta_y|) \geq \frac{\pi}{2} \ \& \ |\mathrm{mod}(\theta_x + \frac{v}{u}\theta_y, 2\pi) - \pi| \leq \frac{\pi}{2}; \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

B. Semi-Discrete Analysis

The SD analysis considers the solutions in the form of $e^{i\theta_y} W_{i_x}, i_x = 0, \ldots, N_x$, where N_x is the number of nodes in the x-direction. The original multidimensional discrete problem is, thus, translated into a one-dimensional problem parametrized by the normalized Fourier frequency, $|\theta_y| \leq \pi$. The discrete function W_{i_x} is either a scalar solution for the convection equation, or a vector solution for the system of flow equations (5). The analysis takes into account specific implementations of boundary conditions and is capable to predict details of solution evolution in individual iterations. When zero manufactured solution is used, the round-off error does not affect computations, which is critical for the ability to observe asymptotic convergence in computations. SD tests routinely encounter and treat residuals as small as 10^{-150}. The asymptotic convergence rate can be directly evaluated as the spectral radius of the iteration matrix. The analysis is precise for a constant-coefficient formulation with y-periodic boundary conditions. A description of the analysis in application to constant-coefficient convection equation is provided in a previous paper.[8]

The DC iteration matrix has the form:

$$\mathbf{DC} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{T}. \tag{11}$$

Here $\mathbf{I}$, $\mathbf{T}$, and $\mathbf{D}$ are the identity, target, and driver matrices, respectively. For the convection equation, $aw_x + bw_y = f$, the matrix $\mathbf{T}$ corresponds to the Fromm discretization, with a row composed of the following coefficients:

$$\mathbf{T} = \left[\begin{array}{ccccccc} \cdots & 0 & \frac{a}{4h_x} & \frac{-5a}{4h_x} & \underline{\frac{3a}{4h_x} + B_2} & \frac{a}{4h_x} & 0 \quad \cdots \end{array} \right], \tag{12}$$

$$B_2 = \frac{b}{4h_y} \left(e^{i\theta_y} + 3 - 5e^{-i\theta_y} + e^{-2i\theta_y} \right), \tag{13}$$

and the main diagonal coefficient is underlined. $\mathbf{D}$ is a driver two-diagonal matrix:

$$\mathbf{D} = \left[\begin{array}{ccccc} \ldots & 0 & \frac{-a}{h_x} & \underline{\frac{a}{h_x} + B_1} & 0 \quad \ldots \end{array} \right], \tag{14}$$

$$B_1 = \frac{b}{h_y} \left(1 - e^{-i\theta_y} \right). \tag{15}$$

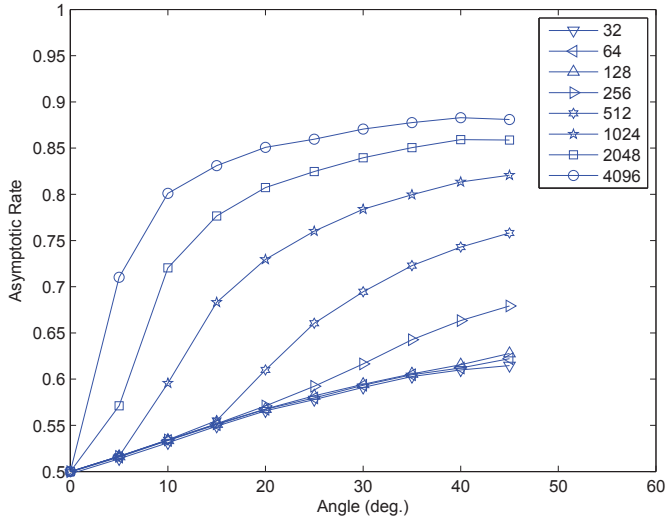
For the system of equations the corresponding matrices are block diagonal.

The iteration matrix of a two-level MG-DC $V(\nu_1, \nu_2)$ cycle is

$$\mathbf{MG} = (\mathbf{DC})^{\nu_2} \mathbf{CGC} (\mathbf{DC})^{\nu_1}, \tag{16}$$

$$\mathbf{CGC} = \mathbf{I} - \mathbf{P}\mathbf{T}_c^{-1}\mathbf{R}\mathbf{T}. \tag{17}$$

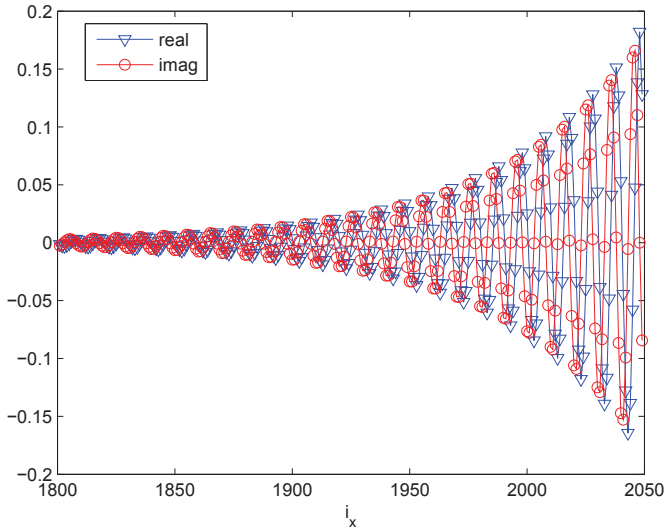
Here $\mathbf{CGC}$ is the coarse-grid-correction matrix, $\mathbf{R}$ and $\mathbf{P}$ are restriction and prolongation matrices, respectively, and $\mathbf{T}_c$ is the coarse-grid-operator matrix. The size of the multigrid matrices is twice as large as the size of corresponding single-grid matrices because multigrid couples two components corresponding to Fourier frequencies θ_y and $\theta_y + \pi$.



□i□□re □□. Asym□t□tic c□n□er□ence rate □□D□ iterati□ns □r c□nstant-c□e□□cient c□n□ecti□n e□□ati□n c□m□□te□ □it□ t□e SD analysis.

□. Scalar c□n□ecti□n e□□ati□n

The asymptotic convergence of DC iterations for the scalar convection equation is computed with the SD analysis. The variation of the asymptotic rate with the grid density and the angle of attack is shown in Fig. 10. The convergence plots on grids of moderate size with up to 128^2 degrees of freedom are practically over-plotted. For small angles of attack, grids with 256^2 and 512^2 degrees of freedom also show similar rates. □n finer grids, however, the convergence rates are dramatically different. Slow asymptotic rates are observed for solutions that are exponentially decaying from the outflow boundary toward the interior. Fig. 11 shows the real and imaginary components of an eigensolution for DC iterations on a grid with $N_x = 2048$ and $\alpha = 45°$ □only variation near the outflow boundary is shown. The eigensolution corresponds to $\theta_y = \frac{\pi}{5}$ and the eigenvalue $\mu = 0.8464 - 0.1382i$.



□i□□re □□. □i□ens□□ti□n □n a □ri□ □it□ $N_x = 2048; \alpha = 45°; \theta_y = \frac{\pi}{5}$; an□t□e c□rres□□n□in□ei□en□al□e is $\mu = 0.8464 - 0.1382i$.

□ven for combinations of grids and solutions with fast asymptotic convergence, many slow DC iterations may be encountered before the asymptotic regime is attained. Algorithmic enhancements are required to accelerate slow iterations preceding the asymptotic convergence and to improve asymptotic convergence, if necessary. Multigrid

addresses both these issues. Convergence of standard full-coarsening multigrid cycles for second-order convection discretizations on high-density grids is limited by the factor 0.75.[19] For the diagonal flow alignment the scheme becomes third-order accurate and the limiting factor is even more severe, 0.875. However, these rates are significantly better than slow-iteration DC rates.

D. Defect-correction iterations for small-perturbation flows

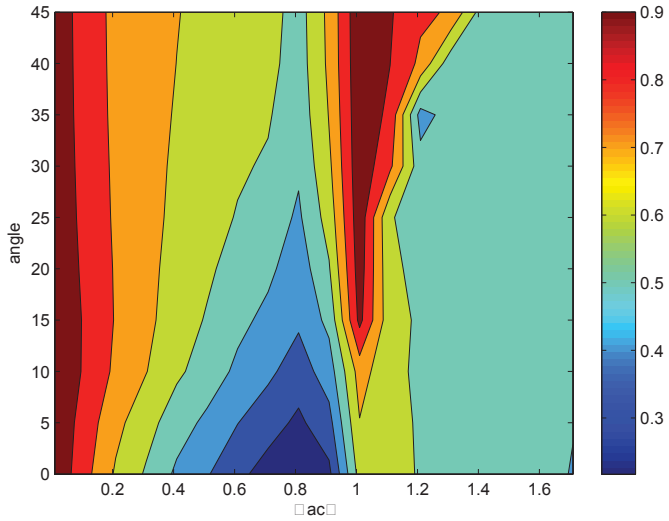


Figure 11. Asymptotic convergence of DC iterations computed with the SD analysis.

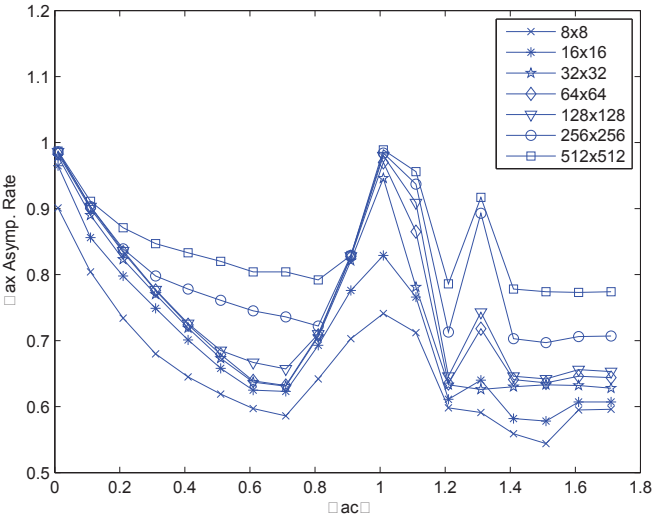
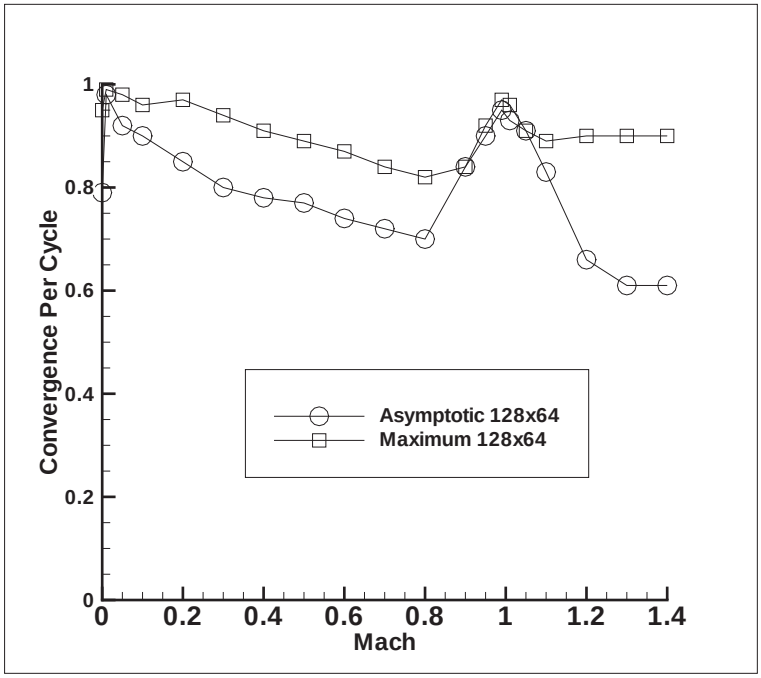


Figure 12. Asymptotic convergence of DC iterations computed with the SD analysis.

In this section, DC iterations are applied for Sflows away from boundaries and singularities. The asymptotic rates of DC iterations are computed with the SD analysis. The angles of attack are varying as $0 \leq \alpha \leq 45°$ and Mach number is varying between (almost) zero and fully supersonic, $0.01 \leq M \leq 1.81$. Fig. 12 shows levels of the asymptotic rate on a 128^2 grid. The grid is not a high-density grid and the rates do not necessarily show the maximum values approached in grid refinement, but the distribution is representative for the medium-density grids. It shows that the slowest convergence is expected at low and near-sonic Mach numbers.

Fig. 13 shows the variation of asymptotic convergence rates versus Mach number on grids of progressively high density. The maximum rate over the range of angles of attack $0 \leq \alpha \leq 45°$ is shown. Slowdown at low and sonic Mach numbers is observed on all grids. Similar to the convection convergence pattern shown in Fig. 10, the rates slow down for all Mach numbers on finer grids.

Actual computations performed on the inflow□outflow domain shown in Fig 1 indicate similar trends. Fig. 14 shows the □asymptotic□rate, namely, the last rate exhibited before achieving the machine-zero error, and the maximum convergence rate observed over the course of iterations. The rates shown in Fig. 14 are somewhat different from the rates predicted by the SD analysis because the error is sometimes reduced to the machine-zero level before the actual asymptotic convergence is achieved. As expected, the maximum rate is closer to one than the asymptotic rate. □oth maximum and asymptotic rates peak at $M \approx 0$ and $M \approx 1$.



□i□re □□. □□n□er□ence rates □□D□ iterati□ns □□ser□e□in act□al S□ □□□ c□m□□tati□ns.

□e□erences

[1] □. Trottenberg, C. W. □osterlee, and A. Schü□er, □Multigrid□, *Academic Press*, □ondon, 2000.

[2] W. A. Mulder, □A high-resolution □uler solver based on multigrid, semi-coarsening, and defect correction□, *Journal of Computational Physics*, 100 (1992), pp. 91□104.

[3] □ Wacker and □. □oren, □Multigrid solution method for the steady □A□S equations□, *Journal of Computational Physics*, 226 (2007), pp. 1784□1807.

[4] W. □. Anderson and D. □. □onhaus, □An implicit upwind algorithm for computing turbulent flows on unstructured grids□, *Computers and Fluids*, 23(1) (1994), pp. 1□21.

[5] F□□3D W□□SIT□. http:□fun3d.larc.nasa.gov. □ast visited □une 2011

[6] □ □. Thomas, D. □. □onhaus, W. □. Anderson, C. □. □umsey, and □. T. □iedron, □An $O(Nm^2)$ plane solver for the compressible □avier-Stokes equations□, AIAA □aper 99-0785, 37-th Aerospace Sciences Meeting □ □xhibit, □eno, □□, □anuary 1999.

[7] □ □. □dwards and □ □. Thomas, □Development of $O(Nm^2)$ preconditioned multigrid solvers for □uler and □avier-Stokes equations□, *AIAA Journal*, 38 (2000), pp. 717□720.

[8] □. Diskin and □ □. Thomas, □Half-space analysis of the defect-correction method for Fromm discretization of convection□, *SIAM J. Sci. Comp.*, 22(2) (2000), pp. 633□655.

[9] H. □ishikawa, □. Diskin, and □ □. Thomas, □Critical study of agglomeration multigrid methods for diffusion□, *AIAA Journal*, 48(4) (2010), pp. 839□847.

[10] □ □. Thomas, □. Diskin, and H. □ishikawa, □A critical study of agglomerated multigrid methods for diffusion on highly-stretched grids□, *Computers and Fluids*, 41(1) (2010), pp. 836□854.

[11] □. Diskin, □ □. Thomas, and □. □. Mineck, □□n quantitative analysis methods for multigrid solutions□, *SIAM J. Sci. Comp.*, 27 (2005), pp. 108□129.

[12] □ □. □oe, □Approximate □iemann solvers, parameter vectors, and difference schemes□, *Journal of Computational Physics*, 43 (1981), pp. 357□372.

[13] J. E. Edwards, "Low-diffusion flux-splitting scheme for Navier-Stokes calculations", *Computers and Fluids*, 26(6) (1998), pp. 635–659.
[14] J. E. Edwards and M. S. Liou, "Low-diffusion flux-splitting methods for flows at all speeds", *AIAA Journal*, 36(9) (1998), pp. 1610–1617.
[15] W. K. Anderson, J. L. Thomas, and B. Van Leer, "Comparison of finite-volume flux vector splittings for the Euler equations", *AIAA Journal*, 24(9) (1986), pp. 1453–1460.
[16] J. L. Thomas, B. Diskin, and C. L. Rumsey, "Towards verification of unstructured-grid solvers", *AIAA Journal*, 46 (2008), pp. 3070–3079.
[17] W. A. Mulder, "A note on the use of symmetric line Gauss-Seidel for the steady upwind differenced Euler equations", *SIAM J. Sci. Stat. Comput.*, 11(2) (1990), pp. 389–397.
[18] A. Brandt, B. Diskin, and J. L. Thomas, "Textbook Multigrid Efficiency for Computational Fluid Dynamics Simulations", AIAA Paper 2001-2570, 15-th AIAA CFD Conference, Anaheim, CA, June 2001.
[19] A. Brandt, "Multigrid techniques: 1984 guide with applications to fluid dynamics", in *Lecture Notes for the Computational Fluid Dynamics*, Lecture Series at the Von-Karman Institute for Fluid Dynamics, 1984.
[20] C. C. Nassberg and A. Jameson, "In pursuit of grid convergence for two-dimensional Euler equations", *Journal of Aircraft*, 47 (2010), pp. 1152–1166.

12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference
10 - 12 September 2008, Victoria, British Columbia Canada

AIAA 2008-5857

Adjoint-based Methodology for Time-Dependent Optimization

N. K. Yamaleev,*B. Diskin,†‡and E. J. Nielsen§

This paper presents a discrete adjoint method for a broad class of time-dependent optimization problems. The time-dependent adjoint equations are derived in terms of the discrete residual of an arbitrary finite volume scheme which approximates unsteady conservation law equations. Although only the 2-D unsteady Euler equations are considered in the present analysis, this time-dependent adjoint method is applicable to the 3-D unsteady Reynolds-averaged Navier-Stokes equations with minor modifications. The discrete adjoint operators involving the derivatives of the discrete residual and the cost functional with respect to the flow variables are computed using a complex-variable approach, which provides discrete consistency and drastically reduces the implementation and debugging cycle. The implementation of the time-dependent adjoint method is validated by comparing the sensitivity derivative with that obtained by forward mode differentiation. Our numerical results show that $O(10)$ optimization iterations of the steepest descent method are needed to reduce the objective functional by 3-6 orders of magnitude for test problems considered.

I. Introduction

Time-dependent optimization problems arise in many areas in science and engineering including various flow control applications such as controlling flow separation, airframe vibration, noise level, transition to turbulence, etc., as well as design optimization problems for essentially unsteady flows, including design and shape optimization of helicopter rotors, turbomachinery blades, aircraft wings, and other configurations. The overall complexity of this class of problems is much higher than that of steady-state aerodynamic optimization problems, which is one of the main reasons why time-dependent optimization has not been used yet in real-life applications. Continuously expanding computer capabilities now attract more attention to numerical solution of time-dependent optimal control and design optimization problems. These problems can be considered as minimization of appropriate cost functionals (e.g., lift, drag, etc.). The resulting control laws or design variables are obtained by solving the corresponding time-dependent optimal control or design problems with appropriate optimization algorithms.

Among various optimization techniques available in the literature (see, e.g., [1]), the adjoint method has recently grown in popularity, rapidly becoming one of the most widely used techniques for solving a variety of steady and unsteady optimization problems. The adjoint methodology is particularly attractive for optimal control/design problems, which include a large number of control variables, yet relatively few constraints. In contrast to a classical forward mode differentiation approach, which requires two flow solves for each control variable, the adjoint methodology has the advantage of computing the cost functional gradients at a fixed expense independent of the number of control/design variables. This property of the gradient methods based on the adjoint formulation make them well suited for steady aerodynamic design optimization problems.[2–5] Although the adjoint-based methods have been successfully used for problems of optimal design within the steady-state aerodynamics, applications of the adjoint formulation to essentially time-dependent optimal control/design problems are still lacking. In,[6] the 2-D continuous time-dependent adjoint incompressible Navier-Stokes equations and optimality conditions have been derived. This continuous adjoint-based method has been successfully used for solving the problem of boundary-layer instability suppression through wave

*Associate Professor, North Carolina A&T State University, Member AIAA. E-mail: nkyamale@ncat.edu
†Senior Research Scientist, National Institute of Aerospace, E-mail: bdiskin@nianet.org
‡Visiting Associate Professor, MAE, University of Virginia, Member AIAA.
§Research Scientist, NASA Langley Research Center, Senior Member AIAA, E-mail: Eric.J.Nielsen@nasa.gov

□□is material is declared a □or□o□t□e □.□. □o□ernment and is not s□□ect to copyrig□t protection in t□e □nited □tates.

cancellation. Nadarajah and Jameson[7] derived and applied the time accurate continuous and discrete adjoint equations to the shape optimization of an oscillating airfoil in an 2-D inviscid transonic flow. In,[8] a gradient method based on the discrete adjoint equations and the corresponding boundary conditions in the frequency domain has been developed. This approach significantly reduces the computational cost for shape optimization of a 3-D wing oscillating at a constant frequency. Note, however, that this technique is applicable only for periodic problems and its efficiency strongly depends on the number of harmonics in the time-dependent solution. Discrete adjoint-based methods operating directly in the time domain have been developed for the 2-D compressible Euler and Navier-Stokes equations in[9] and,[10,11] respectively.

The time-dependent adjoint-based methods mentioned above can be divided into two groups. The first one[7–9,11] is developed for optimization problems with the design/control variables that are independent of time, while the second one[6,10] involves the design/control variables that depend on time. In this paper, we develop a general discrete adjoint-based optimization methodology which is directly applicable to both classes of problems. This time-dependent optimization methodology can be directly applied to solving a very broad spectrum of time-dependent optimal control problems, where the control variables are in general time-dependent (e.g., the displacement of an actuator diaphragm or the velocity distribution at the actuator orifice, etc.) and design optimization problems where the design variable are in general do not depend on time (e.g., shape of a helicopter rotor or an aircraft wing, etc.).

The paper is organized as follows. In Section II, we present the continuous and discrete state equations. In Section III, the discrete time-dependent optimization problem is described. In Section IV, the general discrete time-dependent adjoint equations are derived. Section V discusses a technique for forming discrete adjoint operators by using complex variables. In Section VI, we present two test problems used for validating the developed time-dependent optimization methodology. We draw conclusions and present our plans for the future in Section VII.

II. Governing Equations

We consider the time-dependent, two-dimensional Euler equations describing the unsteady, inviscid compressible flow. The Euler equations written in the integral conservation law form are given by:

$$\frac{\partial V\mathbf{U}}{\partial t} + \oint_{\Gamma} \mathbf{F}\cdot\mathbf{n}d\Gamma = \mathbf{0}, \tag{1}$$

where $\mathbf{n}$ is the outward unit normal vector of the control volume with boundary Γ, V is the control volume, $\mathbf{U}$ is the vector of conserved variables averaged over the control volume, and $\mathbf{F}$ is the Cartesian inviscid flux vector.

The governing equations (1) are discretized by using a node-centered finite-volume scheme, where solution values are stored at the mesh nodes. The control volume around each grid node is constructed by connecting the centroids of the primal-mesh cells with midpoints of the surrounding edges. The discretized Euler equations including the boundary conditions can be written as follows:

$$\frac{\mathbf{Q}^n - \mathbf{Q}^{n-1}}{\Delta t} + \mathbf{R}(\mathbf{Q}^n) = \mathbf{0}, \tag{2}$$

where $\mathbf{Q} = V\mathbf{U}$, and $\mathbf{R}$ is the spatial undivided residual of the discretization, which approximates the contour integral in Eq. (1). It should be noted that the above discrete formulation (2) is very general and can be applied to a broad class of time-dependent PDEs discretized using not only finite volume, but also finite difference and finite element schemes. The flux $\hat{\mathbf{F}}$ in the discretized integral is approximated using Roe's approximate Riemann solver

$$\hat{\mathbf{F}} = \frac{1}{2}\left[\hat{\mathbf{F}}_L + \hat{\mathbf{F}}_R - |\mathbf{A}|\left(\mathbf{U}_L - \mathbf{U}_R\right)\right], \tag{3}$$

where $\hat{\mathbf{F}}_L$ and $\hat{\mathbf{F}}_R$ are the "left" and "right" normal fluxes at the edge midpoint, $\mathbf{U}_L$ and $\mathbf{U}_R$ are the "left" and "right" reconstructed values of the solution vector at the edge midpoint, obtained from some polynomial approximation defined on each control volume, $|\mathbf{A}|$ is the Roe averaged matrix.[12] In Eq. (2), the time derivative has been approximated using the implicit first-order backward-difference (BDF-1) formula. Note that second-order BDF formula as well as higher order implicit Runge-Kutta methods can also be used in the present formulation with minor modifications.

In the present paper, we consider only inviscid flow problems because our primary objective is to develop a time-dependent adjoint-based optimization methodology that is applicable to a broad spectrum of nonlinear state equations. Generalization of this methodology to the unsteady Reynolds-averaged Navier-Stokes (RANS) equations coupled with either one- or two-equation turbulence model is quite straightforward. In this case, only the flux residual $\mathbf{R}$ should be changed, while the adjoint equations, which will be presented in Section IV, remain unchanged. Note, however, that for the RANS equations, questions related to robustness of the present adjoint-based methodology require special investigation, which are outside the scope of the present paper.

III. Discrete Time-Dependent Optimization Problem

We consider the following discrete time-dependent optimization problem:

$$\min_{\mathbf{D}\in\mathcal{D}_a} f(\mathbf{D}), \quad f(\mathbf{D}) = \sum_{n=1}^{N} f^n(\mathbf{D})\Delta t, \quad f^n(\mathbf{D}) = F^n_{\text{obj}}(\mathbf{Q}^n(\mathbf{D})) + F^n_{\text{reg}}(\mathbf{U},\mathbf{D}), \tag{4}$$

where $\mathbf{D}$ is a vector of the control or design variables, which in general depends on time; N is the total number of time steps, over which the control $\mathbf{D}$ is active; $\mathbf{Q}$ is the solution of the unsteady, compressible Euler equations; F^n_{obj} is a part of the cost functional that represents the flow control objective; and F^n_{reg} is a regularization term, typically some weighted norm of the control variable. Note that this setting of the problem remains valid if $\mathbf{D}$ does not depend on time. The above formulation (4) is very general and directly applicable for both time-dependent optimal control problems (e.g., active flow control via synthetic jet actuation) and aerodynamic design optimization for unsteady flows (e.g., design of a turbomachinery blade), and others. The set of admissible controls, $\mathcal{D}_a$, depends on specifics of the target physical system (e.g., how much suction and blowing can the actuators provide, or admissible length and thickness of a blade, etc.), but it should also ensure the existence of a solution of the optimization problem (4).

To reduce the complexity of the optimization problem, without loss of generality, we assume that the objective functional f is a scalar quantity. In the present analysis, we consider the following discrete convex functional:

$$F^n_{\text{obj}} = \beta_1 \left(C^n_L - C^n_{L_{\text{target}}}\right)^2 + \beta_2 \left(C^n_D - C^n_{D_{\text{target}}}\right)^2 \tag{5}$$

where $C^n_{L_{\text{target}}}$ and $C^n_{D_{\text{target}}}$ are given time-dependent target lift and drag coefficients, respectively, which are integrals of the normal and tangential components of the stress tensor over the controlled boundary surface.

The control variables $\mathbf{D}$ have a precise physical meaning (e.g., the Mach number or angle of attack as a function of time, etc.) and should remain bounded and be continuous in time. These physical constraints are incorporated into the optimization problem through the regularization/penalty term in the cost functional Eq. (4), which limits the size of the control. The regularization/penalty term F^n_{reg} is chosen as follows

$$F^n_{\text{reg}} = \frac{\alpha_1}{2}\left[\mathbf{D}^n\right]^T \mathbf{D}^n + \frac{\alpha_2}{2}\frac{1}{\Delta t^2}\left(\mathbf{D}^n - \mathbf{D}^{n-1}\right)^T \left(\mathbf{D}^n - \mathbf{D}^{n-1}\right) \tag{6}$$

where α_1 and α_2 are nonnegative parameters that can be used to adjust the relative weights of the regularization terms appearing in the functional (6). The particular form of the penalty term (6) limits not only the magnitude of the control, but also the rate, at which the control changes, to provide the necessary smoothness of the control. The presence of the second term in the cost functional can also be interpreted as a constraint on the maximum kinetic energy generated by the control system, which is directly related to the energy consumption required for its operation.

It should be noted that the same penalty technique outlined above can be used to impose a more general nonlinear side constraints involving the state variable $\mathbf{U}$. If the optimization problem (4) is subject to the side constraint $\Phi(\mathbf{U},\mathbf{D}) \leq 0$, then the following penalty term can be added to the objective functional to enforce this constraint:

$$F_{\text{reg}} = \alpha \left(\max\left[0, \Phi(\mathbf{U},\mathbf{D})\right]\right)^2, \tag{7}$$

where α is a positive user-defined parameter, and Φ is a continuously differentiable function of its arguments. Note that the above penalty term is continuously differentiable and active only when the constraint is violated, i.e., when $\Phi(\mathbf{U},\mathbf{D}) > 0$. The above penalization guarantees that the constraint $\Phi(\mathbf{U},\mathbf{D}) \leq 0$ is met if $\alpha \to +\infty$. In practice, the parameter α can be increased during the iterative process to make sure that the side constraint is satisfied.

Currently, only matching objective functionals with the zero global minimum have been considered. Because, there are no spurious extrema in all test problems presented herein, the regularization term is set equal to zero. Though this simplified formulation works well for the Euler equations considered in this paper, for problems involving essentially nonlinear one- or two-equation turbulence models, the regularization term may play an important role and should be included into the optimization procedure.

IV. Time-dependent Adjoint Formulation

The discrete time-dependent optimization problem (4) is solved by using the method of Lagrange multipliers which is used to enforce the governing equations and the corresponding boundary conditions (2) as constraints. The discrete Lagrangian functional is defined as follows:

$$L(\mathbf{D},\mathbf{Q},\mathbf{\Lambda}) = \sum_{n=1}^{N} f^n \Delta t \quad + \sum_{n=2}^{N} \left[\mathbf{\Lambda}^n\right]^T \left(\tfrac{\mathbf{Q}^n-\mathbf{Q}^{n-1}}{\Delta t} + \mathbf{R}^n\right)\Delta t + \left[\mathbf{\Lambda}^1\right]^T \left(\mathbf{Q}^1 - \mathbf{Q}^{\text{in}}\right), \tag{8}$$

where $\mathbf{\Lambda}$ is a vector of Lagrange multipliers or costate variables, $n = 1$ corresponds to the initial moment of time, $\mathbf{Q}^{\text{in}}$ is the initial condition for the Euler equations, f^n is the objective functional given by Eq. (5), and $\mathbf{R}^n = \mathbf{R}(\mathbf{Q}^n, \mathbf{D})$ is the spatial undivided residual. Note that the first two terms in the Lagrangian are scaled by Δt, so that they approximate the corresponding time integrals in the continuous Lagrangian. Therefore, the discrete Lagrangian approaches its continuous counterpart as the number of time steps N increases. Furthermore, the scalar product of the costate vector and the vector of the governing equations in Eq. (8) can be interpreted as the integral over the computational domain, which again approximates the continuous Lagrangian.

The sensitivity derivative is obtained by differentiating the Lagrangian with respect to $\mathbf{D}$, which yields

$$\begin{aligned}
\tfrac{dL}{d\mathbf{D}} &= \sum_{n=1}^{N} \left(\tfrac{\partial f^n}{\partial \mathbf{D}} + \left[\tfrac{\partial \mathbf{Q}^n}{\partial \mathbf{D}}\right]^T \tfrac{\partial f^n}{\partial \mathbf{Q}^n}\right)\Delta t + \left[\tfrac{\partial \mathbf{Q}^N}{\partial \mathbf{D}}\right]^T \left(\tfrac{\mathbf{\Lambda}^N}{\Delta t} + \left[\tfrac{\partial \mathbf{R}^N}{\partial \mathbf{Q}^N}\right]^T \mathbf{\Lambda}^N\right)\Delta t \\
&+ \sum_{n=2}^{N-1} \left[\tfrac{\partial \mathbf{Q}^n}{\partial \mathbf{D}}\right]^T \left(\tfrac{\mathbf{\Lambda}^n - \mathbf{\Lambda}^{n+1}}{\Delta t} + \left[\tfrac{\partial \mathbf{R}^n}{\partial \mathbf{Q}^n}\right]^T \mathbf{\Lambda}^n\right)\Delta t \\
&- \left[\tfrac{\partial \mathbf{Q}^1}{\partial \mathbf{D}}\right]^T \Lambda_2 + \left(\left[\tfrac{\partial \mathbf{Q}^1}{\partial \mathbf{D}}\right]^T - \left[\tfrac{\partial \mathbf{Q}^{in}}{\partial \mathbf{D}}\right]^T\right)\mathbf{\Lambda}^1 + \sum_{n=2}^{N} \left[\tfrac{\partial \mathbf{R}^n}{\partial \mathbf{D}}\right]^T \mathbf{\Lambda}^n \Delta t
\end{aligned} \tag{9}$$

Regrouping the terms, Eq. (9) can be recast as follows:

$$\begin{aligned}
\tfrac{dL}{d\mathbf{D}} &= \sum_{n=1}^{N} \tfrac{\partial f^n}{\partial \mathbf{D}}\Delta t + \left[\tfrac{\partial \mathbf{Q}^N}{\partial \mathbf{D}}\right]^T \left(\tfrac{\mathbf{\Lambda}^N}{\Delta t} + \left[\tfrac{\partial \mathbf{R}^N}{\partial \mathbf{Q}^N}\right]^T \mathbf{\Lambda}^N + \tfrac{\partial f^N}{\partial \mathbf{Q}^N}\right)\Delta t \\
&+ \sum_{n=2}^{N-1} \left[\tfrac{\partial \mathbf{Q}^n}{\partial \mathbf{D}}\right]^T \left(\tfrac{\mathbf{\Lambda}^n - \mathbf{\Lambda}^{n+1}}{\Delta t} + \left[\tfrac{\partial \mathbf{R}^n}{\partial \mathbf{Q}^n}\right]^T \mathbf{\Lambda}^n + \tfrac{\partial f^n}{\partial \mathbf{Q}^n}\right)\Delta t \\
&+ \left[\tfrac{\partial \mathbf{Q}^1}{\partial \mathbf{D}}\right]^T \left(\tfrac{\mathbf{\Lambda}^1 - \mathbf{\Lambda}^2}{\Delta t} + \tfrac{\partial f^1}{\partial \mathbf{Q}^1}\right)\Delta t - \left[\tfrac{\partial \mathbf{Q}^{in}}{\partial \mathbf{D}}\right]^T \mathbf{\Lambda}^1 + \sum_{n=2}^{N} \left[\tfrac{\partial \mathbf{R}^n}{\partial \mathbf{D}}\right]^T \mathbf{\Lambda}^n \Delta t
\end{aligned} \tag{10}$$

For problems with a large number of control/design variables, it is desirable to avoid the calculation of $\partial\mathbf{Q}/\partial\mathbf{D}$ in the optimization procedure. Taking into account that so far no constraints have been imposed on the Lagrange multipliers, the $\partial\mathbf{Q}/\partial\mathbf{D}$ term can be eliminated from Eq. (10) by setting the second, third, and forth terms on the right hand side equal to zero, which results in the following adjoint equations for determining the Lagrange multipliers:

$$\frac{1}{\Delta t}\mathbf{\Lambda}^N + \left[\frac{\partial \mathbf{R}^N}{\partial \mathbf{Q}^N}\right]^T \mathbf{\Lambda}^N = -\frac{\partial f^N}{\partial \mathbf{Q}^N} \tag{11}$$

$$\frac{1}{\Delta t}\left(\mathbf{\Lambda}^n - \mathbf{\Lambda}^{n+1}\right) + \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}^n}\right]^T \mathbf{\Lambda}^n = -\frac{\partial f^n}{\partial \mathbf{Q}^n}, \quad 2 \le n \le N-1 \tag{12}$$

$$\frac{1}{\Delta t}\left(\mathbf{\Lambda}^1 - \mathbf{\Lambda}^2\right) = -\frac{\partial f^1}{\partial \mathbf{Q}^1}, \tag{13}$$

Equations (11) and (13) are initial and terminal conditions for the costate variables. Equations (12) represent a linear system of equations for the costate variables, which are solved backward in time. Once Eqs. (11-13)

have been solved, the vector of Lagrange multipliers $\mathbf{\Lambda}^n$ can be used to evaluate the last two terms in Eq. (10). As a result, the sensitivity derivative can be calculated as follows:

$$\frac{dL}{d\mathbf{D}} = \sum_{n=1}^{N} \frac{\partial f^n}{\partial \mathbf{D}} \Delta t + \sum_{n=2}^{N} \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{D}} \right]^T \mathbf{\Lambda}^n \Delta t - \left(\frac{\partial \mathbf{Q}^{\text{in}}}{\partial \mathbf{D}} \right)^T \mathbf{\Lambda}^1, \tag{14}$$

where $\frac{\partial f^n}{\partial \mathbf{D}}$ and $\frac{\partial \mathbf{R}^n}{\partial \mathbf{D}}$ are calculated by using $\mathbf{Q}^n$ stored during the forward sweep in time.

The minimum of the functional is found by using the steepest descent method in which each step of the optimization cycle is taken in the negative gradient direction

$$D_m^{(k+1)} = D_m^{(k)} - \tau_m \frac{dL}{dD_m}, \tag{15}$$

where τ_m is the step size for D_m, which is the m-th component of the vector $\mathbf{D}$, and k is the optimization cycle counter. The sensitivity derivative dL/dD_m in Eq. (15) is determined using Eq. (14) which requires the solution of the adjoint equations (11-13). During the solution of the adjoint equations that are integrated backward in time, the sensitivity derivative at each time step is computed and added to its value at the previous time step. At $n = 1$, the complete sensitivity derivative is available and used in Eq. (15) for calculating a new value of the control variable $\mathbf{D}^{(k+1)}$. Then the entire optimization cycle is repeated until $|L^{(k+1)} - L^{(k)}| < \epsilon$, where ϵ is a given tolerance. This optimization algorithm has been selected because of its simplicity; it is known to be sensitive to the step size τ. Other more efficient and robust gradient-based methods, such as conjugate gradient or quasi-Newton methods, can also be easily coupled with the time-adjoint formulation used in the present analysis.

V. Forming Discrete Adjoint Operators by Using Complex Variables

As follows from Eqs. (11-13), the derivatives of $\mathbf{R}$ and f with respect to $\mathbf{Q}$ and $\mathbf{D}$ are required to form the adjoint equations and the sensitivity derivative. It is very difficult and time-consuming to obtain these derivatives by manually differentiating a CFD solver, especially if complicated turbulence and physical models are involved. Furthermore, any changes in the discretization of the governing equations, boundary conditions, objective functional, or physical models require additional coding and debugging, thus making the software development cycle extremely lengthy. To overcome these difficulties, we use an approach based on complex variables, which has successfully been applied to solving design optimization problems in [13, 14]. The key idea of this technique is to approximate the required real-valued derivatives by using the following formula proposed by Lyness[15] :

$$\frac{\partial f}{\partial x} = \frac{Im\left[f(x + ih)\right]}{h} + O(h^2), \tag{16}$$

where $f(x)$ is a complex-valued function. In contrast to the finite-difference method, the above complex variable formula is robust for small h, while providing true second order accuracy. Another advantage of this approach is that no additional flow solves are required to evaluate this derivatives, because only the solution at the current time level $\mathbf{Q}^n$ is needed to compute $\mathbf{R}$ and f and their perturbed values. The complex variable approach drastically reduces the implementation cycle and provides adjoint-based optimization capabilities for realistic physical and turbulence models. Note, however, that this approach is not without penalties in the CPU time and memory as compared with the handcoded Jacobians implementation because complex arithmetic is used.

VI. Numerical Results

In this section, we present computational results demonstrating how the adjoint-based method performs for two time-dependent optimization problems, involving flow matching functionals. The key distinction between these two problems is that for the first one, a design/control variable is independent of time, while for the second problem, control variables depend on time.

The first problem is a minimization of a matching functional given by Eq. (5) with $\beta_2 = 0$ for the unsteady flow around a bump. The unsteadiness is introduced into the flow through the freestream Mach number, which oscillates in time

$$M(t) = M_0 + \Delta M \cos(\omega t), \tag{17}$$

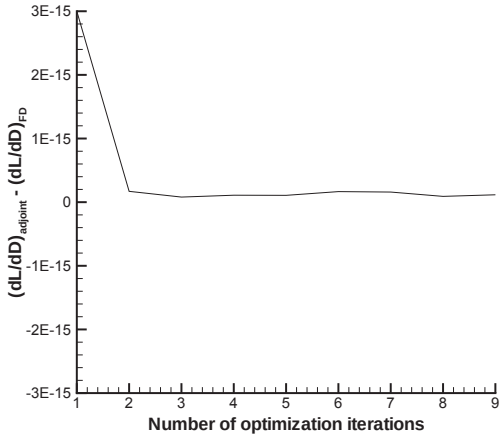


Figure 1. Comparison of the sensitivity derivatives computed using the finite difference and adjoint methods for the first test problem.

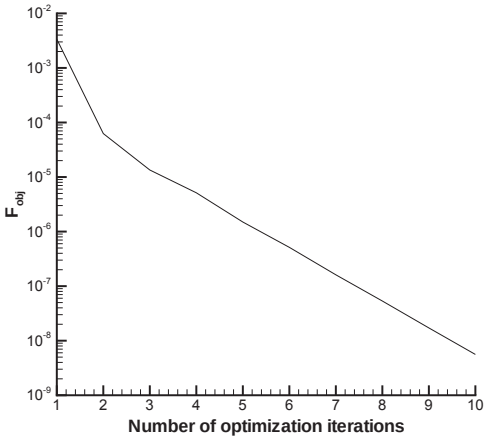


Figure 2. Convergence history of the objective functional $F_{\rm obj}$.

where M_0 is a mean value of the freestream Mach number, ΔM is a Mach number amplitude, and ω is a frequency of Mach number oscillations. The thickness of the bump is set to be 10% of its chord length. The flow conditions used in this test problem are: $M_0 = 2.0$, $\omega = 17\pi/9$, and $T_{\rm final} = 1$. For this test problem, the Mach number amplitude ΔM is used as a control variable. Note that this control variable is independent of time. The time-dependent target lift coefficient $C_{L_{\rm target}}(t)$ in the objective functional is calculated numerically by solving the unsteady Euler equations (2) with $\Delta M = 0.5$. A solution obtained at $\Delta M = 0.1$ is used as a starting point for the optimization procedure. The optimization is stopped when the absolute value of the difference between the current value of the Lagrangian and its value at the previous optimization iteration is less than 10^{-8}. This test problem is solved on a 61×21 structured grid using a node-centered finite volume code that is first-order accurate both in time and space. At each time step, the nonlinear discrete flow equations are solved by using the Newton's method. The adjoint equations are integrated backward in time and require the solution of the Euler equations to be known at all time steps, over which the optimization problem is solved. In the present implementation, the entire unsteady solution set is held in operating memory. For mid-size 2D problems integrated over $O(10^2)$ time steps, the required solution history, can be stored on a hard drive, as has been reported in [9]. In this case, the speed of I/O operation itself does not have a significant effect on the overall CPU time. Note, however, that for realistic

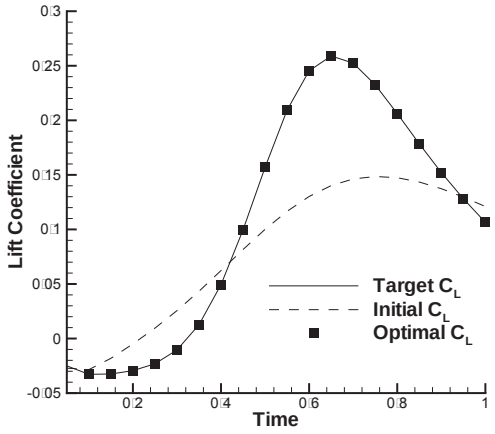


Figure 3. Comparison of the optimal lift coefficient computed using the adjoint-based method with its initial and target values for the first test problem.

3D nonperiodic problems, this approach can quickly become prohibitive in terms of the disk memory and the CPU time, so more efficient approaches are needed to solve this class of unsteady optimization problems.

To evaluate the accuracy of calculation of $dL/d\mathbf{D}$ and to check the implementation of the adjoint solver, two different methods for computing the sensitivity derivative are used and compared with each other. The first method is based on a forward mode differentiation of the Lagrangian with respect to the control variable, which is implemented by using the complex variable approach (16). The second method uses the discrete adjoint formulation described in Section IV. Note that at each optimization cycle, for the first method, the flow problem should be solved twice for each control variable. In contrast to the first method, the second approach requires a single solution of the Euler and corresponding adjoint equations per optimization cycle, regardless of the number of the control/design variables. For the forward mode differentiation, the complex step size has been chosen to be 10^{-7}. Figure 1 shows the difference between the sensitivity derivatives obtained using the finite difference and adjoint methods. As seen in the figure, the discrepancy is of the order of the round-off error, thus corroborating the validity of the time-dependent adjoint formulation. The history of convergence of the objective functional is presented in Fig. 2. The value of the objective functional drops by an order of magnitude every 2 optimization cycles. Note that this convergence behavior remains practically unchanged until the functional becomes smaller than the specified tolerance when the optimization was stopped. Only 10 optimization iterations were needed to reduce the objective functional by six orders of magnitude. To illustrate that the lift coefficient converges to its target value, time histories of the optimal, target, and initial lift coefficients are depicted in Figure 3. At the first optimization iteration, the maximum value of the time-dependent lift coefficient is about two times less that that of the target $C_L(t)$. After 10 optimization iterations, the time history of the computed lift coefficient is practically indistinguishable from the target solution at all time steps.

The second test problem is similar to the first one, but now, values of the freestream Mach number at each time step $\mathbf{D} = (M_1, \ldots, M_N)^T$ are used as control variables. The optimization procedure starts at $M = 2.1$ which is used as an initial guess. The objective functional for this test problem is given by

$$F_{obj} = \sum_{\Gamma_c} \left(P_j^n - P_{\text{target}}^n\right)^2 \ ,$$

where P_j^n and P_{target}^n are computed and target time-dependent pressure profiles at the lower boundary of the computational domain. The target pressure distribution is calculated numerically by solving the same unsteady problem with $M = 2 + 0.5\cos(17\pi t/9)$. The optimization is stopped when either the relative change in the value of each control variable becomes smaller than 10^{-4} or the absolute value of the objective functional becomes smaller than 10^{-7}. As in the previous case, the adjoint equations are integrated backward in time and require the solution of the Euler equations to be known at all time steps, which is held in the operating memory. The selection of the optimization step size τ in Eq. (15) has a strong effect on the

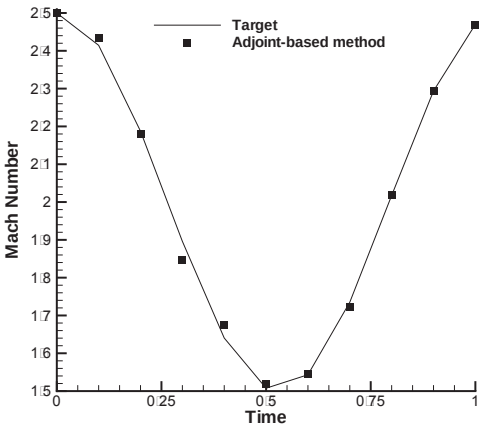


Figure 4. Comparison of the target Mach number and the optimal Mach number computed using the adjoint-based method for the second test problem.

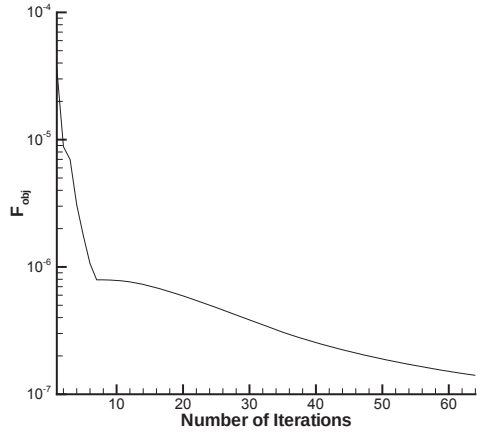


Figure 5. A history of convergence of the adjoint-based optimization method for the second test problem.

number of optimization iterations required to reach an optimum solution. Large step sizes may result in instabilities in optimization iterations, whereas small step sizes provide stability, but drastically slow down the convergence. Therefore, for this test problem, the optimization step size in Eq. (15) is selected adaptively to maximize the convergence rate. In the present analysis, the following algorithm for choosing τ is used:

1) Each optimization cycle continues until a solution with a smaller integrated cost functional, $\sum_{n=1}^{N} f^n \Delta t$, has been found.
2) For a trial vector of τ, the new vector $\mathbf{D}$ and the corresponding $\mathbf{U}$ are computed on all time levels.
3) The n-th component of the vector τ, τ_n, is changed if one of the following two events occurs:
4) if the local cost functional increases, i.e., $f^n_{\text{trial}} > f^n$, then τ_n is decreased, $\tau_n = 0.5\tau_n$.
5) or if the local cost functional decreases slowly, $f^n > f^n_{\text{trial}} > 0.9 f^n$,
   then τ_n is changed depending on signs of $\frac{dL}{dD_n}$ at the current and the previous optimization cycles.
6) if signs are opposite, i.e., $\left(\frac{dL}{dD_n}\right)^{\text{current}} \left(\frac{dL}{dD_n}\right)^{\text{previous}} < 0$, then τ_n is decreased, $\tau_n = 0.5\tau_n$,

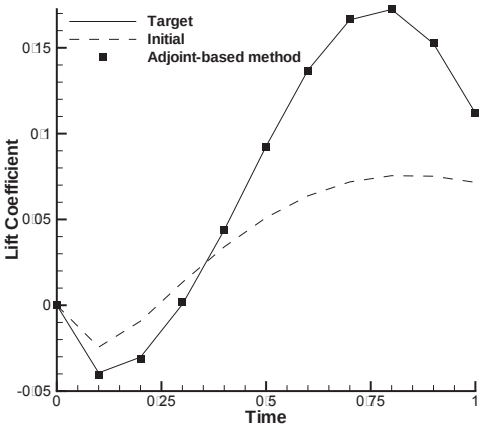


Figure 6. Comparison of the optimal lift coefficient computed using the adjoint-based method with its initial and target values for the second test problem.

7) otherwise, the signs are the same and τ_n is increased, $\tau_n = 1.5\tau_n$.

Here, f^n_{trial} and f^n are the trial and current values of the objective functional at the time level n, and dL/dD_n is the sensitivity derivative with respect to the control variable D_n.

Figure 4 shows the optimal and target Mach number distributions in time. As seen in the figure, the time-dependent optimization method converges to the target solution on the entire time interval considered, thus validating the unsteady adjoint formulation.

A history of convergence of the objective functional obtained with adjoint-based optimization technique is presented in Fig. 5. The total number of optimization cycles required for the adjoint-based optimization method to converge is an order of magnitude larger than that obtained for the first test problem. This is not surprising, because the dimensionality of the design space has also increased by an order of magnitude.

To illustrate that the lift coefficient converges to its target value, the optimal C_L obtained with time-dependent optimization method as well as the initial and target lift coefficients are presented in Figure 6. The relative difference between the initial lift coefficient and its target value is of the order of $O(1)$, while the solution obtained with the adjoint-based method is almost indistinguishable from the target C_L over the entire time interval considered.

VII. Conclusions

We have developed the general adjoint-based methodology for solving a broad spectrum of optimal flow control and design optimization problems. The methodology is directly applicable to both time-dependent optimization problems with control/design variables that are time-dependent and design optimization problems with the control variables that do not depend on time. Nonlinear constraints on the control/design and state variables can be incorporated into the present formulation by introducing the penalty/regularization term in the cost functional. The discrete adjoint operators required for this formulation are computed by using the complex variable approach which is robust for very small step sizes, thus providing adjoint-based optimization capabilities for realistic physical models. The present adjoint-based methodology has been validated using two test problems involving flow matching functionals. Applications of this adjoint-based methodology to more realistic time-dependent design optimization problems involving moving and deforming grids is currently under investigation. Our future research will also focus on developing optimization and computational techniques for reducing the CPU and memory cost of the present time-dependent adjoint-based methodology.

References

[1]Borggaard, J., Burkardt, J., Gunzburger, M., and Peterson, J., Optimal design and control, Boston: Birkhauser, 1995.

[2]Jameson, A., Pierce, N., Martinelli, L., "Optimum Aerodynamic Design Using the Navier-Stokes Equations," *Theor. Comput. Fluid Dyn.* Vol. 10, No. 1, pp. 213–237, 1998.

[3]Anderson, W.K., Bonhaus, D.L., "Airfoil Design on Unstructured Grids for Turbulent Flows," *AIAA J.* Vol. 37, No. 2, pp. 185–191, 1998.

[4]Nielsen, E.J., and Anderson, W.K., "Recent Improvements in Aerodynamic Optimization on Unstructured Meshes," *AIAA J.* Vol. 40, No. 6, pp. 1155–1163, 2002.

[5]Soto, R., and Yang, C., "An Adjoint-based Methodology for CFD Optimization Problems," *AIAA Paper* 2003-0299, 2003.

[6]Joslin, R.D, Gunzburger, M.D., Nicolaides, R.A., Erlebacher, G., and Hussaini, M.Y., "Self-Contained Automated Methodology for Optimal Control," *AIAA J.* Vol. 35, No. 5, pp. 816–824, 1997.

[7]Nadarajah, S.K., Jameson, A., "Optimal Control of Unsteady Flows Using a Time Accurate Method," *AIAA Paper* 2002-5436, 2002.

[8]Nadarajah, S.K., McMullen, M., and Jameson, A., "Non-Linear Frequency Domain Based Optimum Shape Design for Unsteady Three-Dimensional Flow," *AIAA Paper* 2006-1052, 2006.

[9]Mani, K., Mavriplis, D. J., "An Unsteady Discrete Adjoint Formulation for Two-Dimensional Flow Problems with Deforming Meshes," *AIAA Paper* 2007-60, 2007.

[10]Collis, S. S., Ghayour, K., Heinkenschloss, M., Ulbrich, M., Ulbrich, S., "Optimal Control of Unsteady Compressible Viscous Flows," *Int. J. Numer. Meth. Fluids*, Vol.40, No. 11, 2002, pp. 1401-1429.

[11]Rumpfkeil, M.P., and Zingg, D.W., "A General Framework for the Optimal Control of Unsteady Flows with Applications," *AIAA paper*, 2007-1128.

[12]Roe P. L., "Approximate Riemann solvers, parameter vectors, and difference schemes", *J. Comp. Phys.*, vol. 43, 1981, pp. 357-372.

[13]Anderson, W. K., Newman, J. C., Whitfield, D. L., and Nielsen E. J., "Sensitivity Analysis for the Navier-Stokes Equations on Unstructured Meshes Using Complex Variables, *AIAA Journal*, Vol. 39, No. 1, 2001, pp. 56–63.

[14]Nielsen, E. J., and Kleb, W. L., "Efficient Construction of Discrete Adjoint Operators on Unstructured Grids by Using Complex Variables, *AIAA Journal*, Vol. 44, No. 4, 2005, pp. 827–836.

[15]Lyness, J. N., "Numerical Algorithms Based on the Theory of Complex Variables," Proc. ACM 22$^{\text{nd}}$ Nat. Conf., Thomas Book Co., Washington, D.C., 1967, pp. 124–134.